NCT® 99M

NCT® 2000M

Comenzi numerice pentru maşini de frezat
şi centre de prelucrare

Manualul programatorului

Produs al firmei **NCT Automation kft.**
H1148 Budapest Fogarasi no. 5 -7
Mail : 1636 Bp.pf. : 26
Telefon: (+36 1) 467 63 00
Fax : (+36 1) 363 66 05
E-mail: nct@nct.hu
Internet: http://www.nct.hu

Cuprins

27 Noiembrie 2001

1 Introducere

1.1 Programul piesă

Programul piesă este un set de instrucțiuni care pot fi interpretate de către comanda numerică pentru a putea controla operarea maşinii.
Programul piesă constă din blocuri, care sunt compuse din cuvinte.

Cuvânt: Adresă şi Dată
Fiecare cuvânt este alcătuit din două părți – o adresă şi o dată. Adresa are unul sau mai multe caractere, data este o valoare numerică (un întreg sau o valoare zecimală). Anumitor adrese li se poate da un semn sau un operator I.

Lista de adrese:

Adresa	Semnificație	Limitele valorii
O	număr program	0001 - 9999
/	bloc opțional	1 - 9
N	numărul blocului	1 - 99999
G	funcție pregătitoare	*
X, Y, Z, U, W	lungimile coordonatelor	I, -, *
A, B, C	coordonate unghiulare, lungimile coordonatelor, funcții auxiliare	I, -, *
R	rază cerc, dată auxiliară	I, -, *
I, J, K	coordonatele centrului cercului, coordonată auxiliară	-, *
E	coordonată auxiliară	-, *
F	viteză de avans	*
S	viteză de rotire arbore	*
M	funcții diverse	1 – 999
T	număr sculă	1 – 9999
H, D	numărul celulei de compensare lungime şi rază	1-99
L	număr de repetări	1 - 9999
P	dată auxiliară, timp de întârziere	-, *
Q	dată auxiliară	-, *
,C	distanță de teşire	-, *
,R	raza de filetare	-, *
,A	unghiul în linie dreaptă	-, *
(	comentariu	*

La o adresă marcată cu * în coloana **Limitele valorii**, data poate avea şi o valoare cu zecimale.
La o adresă marcată cu **I** şi **-** , se poate introduce fie un operator incremental fie un semn.

Semnul pentru valoare pozitivă „+" nu este indicat şi nici nu se memorează.

Bloc (linie de program)

Un bloc este compus din cuvinte.
Blocurile sunt separate în memorie prin caractere ▫▫ (**L**ine **F**eed = avans linie). Folosirea unui număr de bloc nu este obligatorie. Pentru a distinge pe ecran sfârşitul unui bloc de începutul altui bloc, fiecare bloc nou va începe pe o altă linie, cu caracterul **>** plasat la începutul său; în cazul unui bloc mai lung decât un rând, în fiecare rând nou cuvintele vor începe cu caracterul linie.

Număr de program şi Nume de program

Numărul programului şi numele programului sunt folosite pentru a identifica programul.
Folosirea unui număr de program este obligatorie, cea a numelui de program nu este.
Adresa unui **număr de program** este O. Ea trebuie să fie urmată de exact **patru cifre**.
Numele programului este o secvenţă arbitrară de caractere (şir) puse între semnele de deschidere şi de închidere paranteze **( )** . Poate avea maxim 16 caractere.
Numărul programului şi numele programului sunt separate prin caractere ▫▫▫(**L**ine **F**eed = avans linie) de celelalte blocuri de program aflate în memorie.
În cursul editării, numărul programului şi numele programului vor fi afişate invariabil în prima linie.
În memorie nu pot exista două programe cu acelaşi număr.

Început de program şi Sfârşit de program

Fiecare program începe şi se sfârşeşte cu caracterele %. În cursul editării programului piesă, caracterul de terminare a programului este plasat invariabil după ultimul bloc pentru a se asigura că chiar şi în cazul unei căderi a alimentării cu energie electrică pe timpul editării, se păstrează condiţia de terminare a programului.

Formatul programului în memorie

Programul stocat în memorie este un set de caractere ASCII.
Formatul programului este:

```
%O1234( NUME PROGRAM)▫▫▫ /1N12345G1X0Y...▫▫▫ G2Z5...▫▫▫....▫▫▫
...▫▫▫
...▫▫▫
N1G40...M2▫▫▫
%
```

În secvenţa de caractere de mai sus,
▫▫▫ este caracterul „avans de linie",
% este începutul (şi sfârşitul) de program.

Formatul programului în comunicaţia cu dispozitivele externe

Cele spuse mai sus se aplică şi în cazul comunicaţiei cu un dispozitiv extern.

Program principal şi subprogram

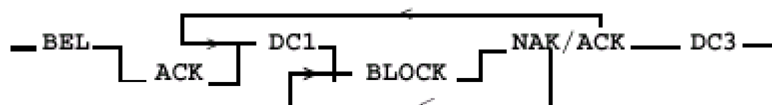
Programele piesă pot fi împărţite în două grupe principale –
programe principale, şi
subprograme.

Procedura de prelucrare a unei piese este descrisă în programul principal. Dacă pe parcursul prelucrării trebuie executate aceleaşi operații în diverse locuri, nu este necesar ca acele secțiuni de program care descriu operațiile respective să fie scrise de mai multe ori în programul principal, ci trebuie scris un subprogram, care poate fi apelat din orice loc (chiar şi din alt subprogram). Utilizatorul poate reveni din subprogram în programul care a făcut apelul.

Canal DNC

Se poate executa şi un program care se află memorat pe o unitate externă (de exemplu pe un calculator PC), fără ca programul să se memoreze în comanda numerică. În acest caz comanda numerică în loc să citească programul din memoria proprie îl va citi din mediul de stocare extern prin intermediul interfeței RS232C. Se face referire la această legătură ca fiind „Canal DNC". Această metodă este folositoare îndeosebi atunci când se execută programe prea mari pentru a fi conținute în memoria comenzii numerice.
Canalul DNC este un canal de transfer de date controlat cu protocol, după cum se arată mai jos.

Comandă:
Echipament:



Prescurtările de mai sus au următoarele semnificații (şi codurile lor ASCII):

BEL (7): Comanda cere emițătorului să stabilească comunicația. Comanda trimite BEL din nou dacă nu primeşte ACK într-o perioadă definită de timp.
ACK (6): Confirmare.
NAK (21): Transfer de date eronat (de exemplu defecțiune hardware pe linie sau eroare BCC). Transferul BLOCULUI trebuie repetat.
DC1 (17): Trebuie pornit transferul următorului BLOC.
DC3 (19): Întreruperea comunicării.
BLOCK:
- În principal un bloc NC (incluzând caracterul de terminare linie ▫▫) şi suma de control până în acel loc (BCC) stocată pe 7 biți ca ultimul byte al blocului (bitul 7, cel mai semnificativ, al BCC este întotdeauna 0). În bloc nu poate fi conținut nici un caracter ASCII cu cod mai jos decât SPACE (32) inclusiv.
- **EOF** (26) (**E**nd **O**f **F**ile = sfârşit fişier), un semnal este transferat de către echipament (emițător) pentru a întrerupe comunicația.

Pentru modul DNC, se impune pentru cel de al doilea canal fizic (numai acesta se poate folosi drept canal DNC) modul 8 biți cu paritate pară.
Un program principal executat din canalul DNC poate avea doar o secvență liniară. Aceasta nu se aplică la subprograme sau macro (dacă a fost apelat vreunul), totuşi acestea trebuie să fie conținute în memoria comenzii numerice. În cazul unei ramificații de la secvența liniară în programul principal (GOTO, DO WHILE), comanda numerică va transmite mesajul „3058 NOT IN DNC" (3058 Nu se află în DNC). Dacă comanda numerică detectează o eroare de BLOCK şi declară NAK, trebuie să fie repetat BLOCUL.

1.2 Termeni fundamentali

Interpolarea

Sistemul de control poate deplasa scula în timpul prelucrării de-a lungul unor linii drepte sau a unor arce. În continuare se va face referire la aceste activități ca fiind „interpolare“.

Mişcarea sculei pe o linie dreaptă:

Program:

```
G01 Z___
 X__ Z___
```

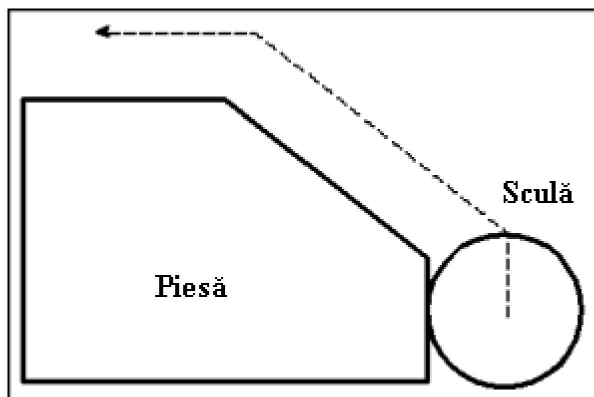


Fig. 1.2-1

Mişcarea sculei pe un arc:

Program:

```
G02 X___ Z___ R___
```

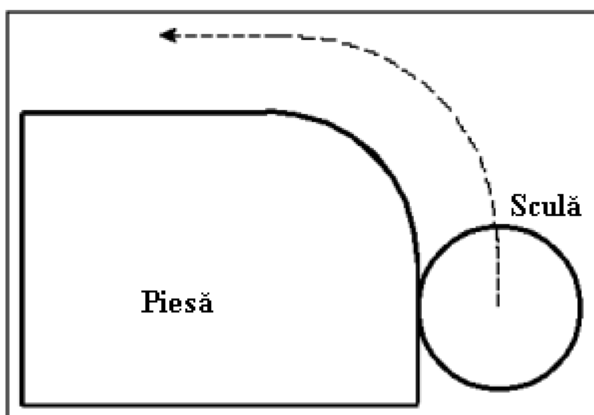


Fig. 1.2-2

Funcții pregătitoare (coduri G)

Tipul de activitate care urmează să fie executat de către un bloc este descris prin folosirea funcțiilor pregătitoare (la care se face referire ca fiind codurile G). De exemplu codul G01 introduce o interpolare liniară.

Avans

Termenul „avans“ se referă la viteza sculei relativă la piesa de prelucrat pe durata procesului de tăiere. Avansul dorit se poate specifica în program la adresa F şi cu o valoare numerică. De exemplu F150 semnifică 150 mm/min.

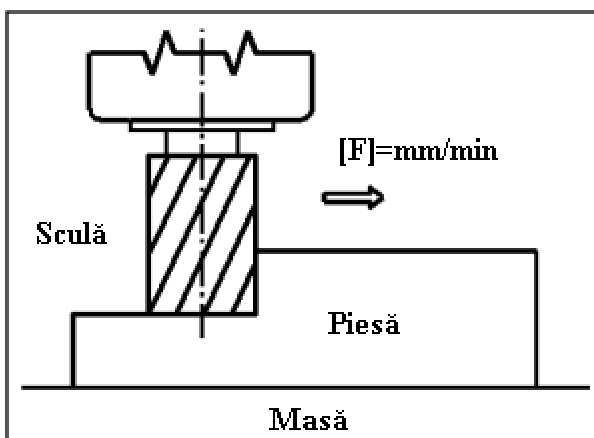


Fig. 1.2-3

Punct de referință

Punctul de referință este un punct fix de pe maşina unealtă. După cuplarea la rețea a maşinii,

axele trebuie să fie deplasate în punctul de referință. După aceea sistemul de control va fi capabil să interpreteze şi datele de coordonate absolute.

Sistem de coordonate

Dimensiunile indicate în desenul piesei sunt măsurate dintr-un anumit punct al piesei. Acel punct este originea sistemului de coordonate al piesei. Aceste date dimensionale trebuie să fie scrise la adresa coordonatei în programul piesă. De exemplu X340 înseamnă un punct cu coordonatele 340 mm în sistemul de coordonate al piesei.

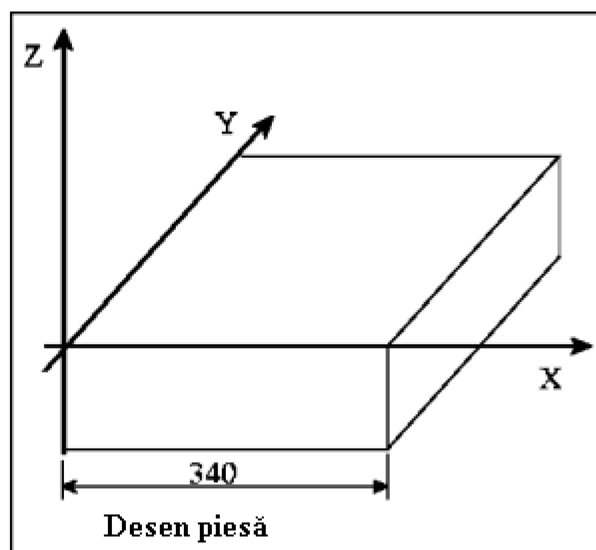


Fig. 1.2-4

Sistemul de coordonate specificat în care comanda numerică interpretează pozițiile este diferit de sistemul de coordonate al piesei. Pentru executarea unei piese corecte, trebuie impusă aceeaşi origine pentru cele două sisteme de coordonate în comanda numerică. Aceasta se poate face de exemplu prin deplasarea centrului sculei într-un punct al piesei cu poziție cunoscută şi impunerea sistemului de coordonate al comenzii numerice la acea valoare.

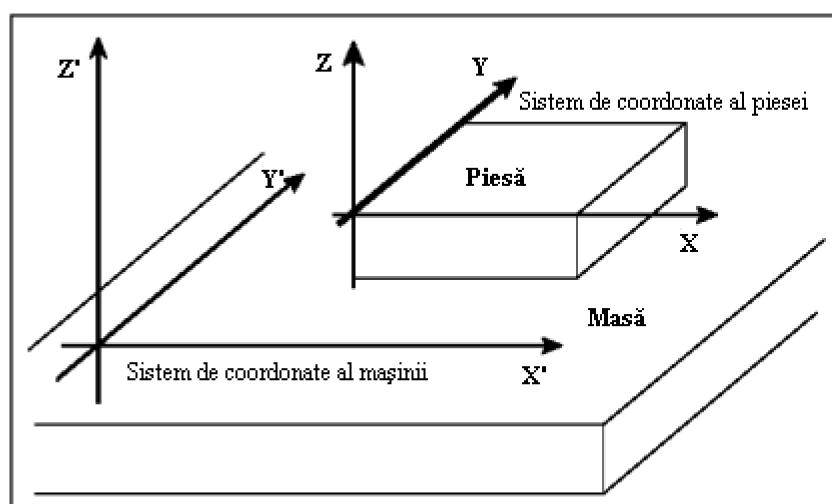


Fig. 1.2-5

Specificarea coordonatei absolute

Când sunt specificate coordonate absolute, scula se deplasează pe o distanţă măsurată de la originea sistemului de coordonate, adică într-un punct a cărui poziţie a fost specificată prin coordonate.

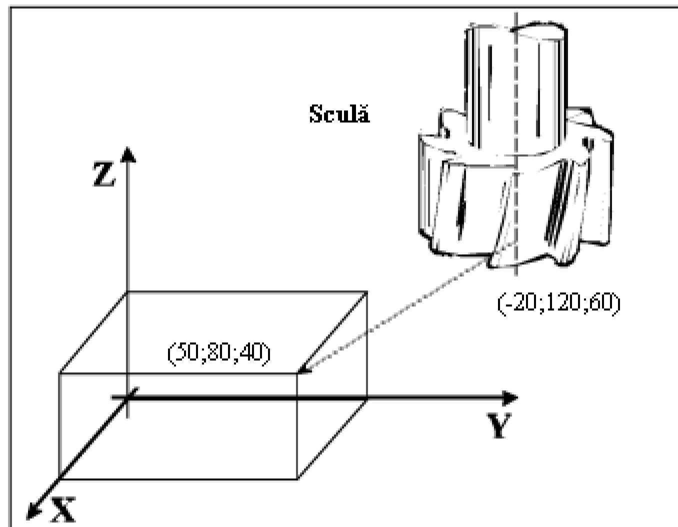


Fig. 1.2-6

Codul specificării de dată absolută este G90.
Blocul

```
G90 X50 Y80 Z40
```

va deplasa scula în punctul din poziţia de mai sus, indiferent de poziţia sa înainte de a se da această comandă.

Specificarea coordonatei incrementale

În cazul unei specificări incrementale a datei, sistemul de control va interpreta coordonata astfel încât scula se va deplasa cu o distanţă măsurată din punctul în care se află în acel moment.
Codul specificării de dată incrementale este G91. Codul G91 se referă la toate valorile coordonatei.
Blocul:

```
G91 X70 Y-40 Z-20
```

va deplasa scula pe distanţa respectivă începând din poziţia anterioară.

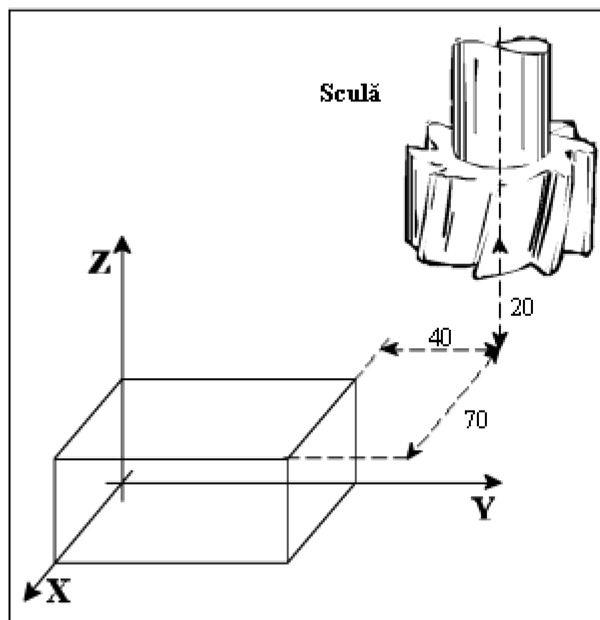


Fig. 1.2-7

O dată incrementală poate fi definită şi ca referindu-se doar la o singură coordonată. Pentru aceasta se introduce caracterul I după adresa coordonatei.
În blocul:

```
G90 XI-70 Y80 Z40
```

data pentru X va fi interpretată ca o valoare incrementală, în timp ce datele pentru Y şi Z sunt interpretate ca şi coordonate absolute pentru codul G90.

Funcţiile modale

Anumite coduri sunt active până când este specificat un alt cod sau valoare. Acestea sunt **coduri modale**. De exemplu, în secvenţa de program:

```
N15 G90 G1 X20 Y30 F180
```

```
N16 X30
N17 Y100
```

codul G90 (specificare de dată absolută) şi valoarea F (avans), specificate în blocul N15, vor fi modale în blocurile N16 şi N17. Astfel nu mai este necesar să se specifice aceste funcții în fiecare dintre blocurile următoare.

Funcțiile cu o singură execuție (nemodale)

Anumite coduri sau valori sunt active doar în blocul în care sunt specificate. Acestea sunt funcții cu o singură execuție.

Comanda vitezei arborelui

Viteza arborelui poate fi specificată la adresa S. Este denumită şi „funcția S". Instrucțiunea S1500 impune o rotire a arborelui la viteza de 1500 rot/min.

Funcția Schimbare sculă

În decursul prelucrării, pentru operațiile executate trebuie folosite diverse scule. Sculele sunt diferențiate prin numere. Se face referire la scule folosind codul T. Instrucțiunea T25 în program, înseamnă că trebuie schimbată scula nr.25. Schimbarea sculei se poate face manual sau automat, funcție de construcția maşinii.

Funcții diverse

În decursul prelucrării trebuie executate un număr de operații de pornire sau oprire. De exemplu pornirea arborelui principal sau pornirea lichidului de răcire. Aceste operații pot fi executate cu funcțiile M (diverse). Ca exemplu în şirul de instrucțiuni

```
M3 M8
```

M3 semnifică „rotire arbore în sens orar" iar M8 semnifică „pornirea lichidului de răcire".

Compensarea lungimii sculei

În cursul prelucrării sunt folosite scule de diferite lungimi pentru operații diferite. Pe de altă parte, în producția de serie, o anumită operație se poate executa cu scule de lungimi diferite (de exemplu când se rupe scula). În scopul de a face mişcările descrise în programul piesă independente de lungimea sculei, trebuie impuse în comanda numerică diversele lungimi ale sculelor. Dacă prin program se impune deplasarea vârfului sculei într-un punct specificat, trebuie să fie apelată valoarea lungimii sculei respective. Aceasta se face la adresa H. De exemplu instrucțiunea H1 face referire la data de lungime nr.1. Apoi comanda numerică deplasează vârful sculei în punctul specificat. Această procedură este referită ca impunere a modului „compensare a lungimii sculei”.

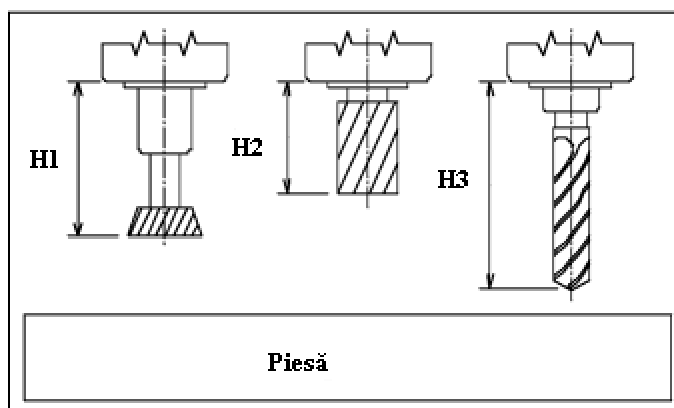


Fig. 1.2-8

Compensarea razei sculei

Prelucrarea unei piese trebuie executată cu scule de raze diferite. Compensarea razei trebuie introdusă pentru a scrie datele conturului real în programul piesă, şi nu pe cele ale traiectoriei centrului sculei (luând în considerare razele sculelor). Valorile compensării razei trebuie introduse în comanda numerică. În continuare, în program, se poate face referire la compensarea razei sculei la adresa D din program.

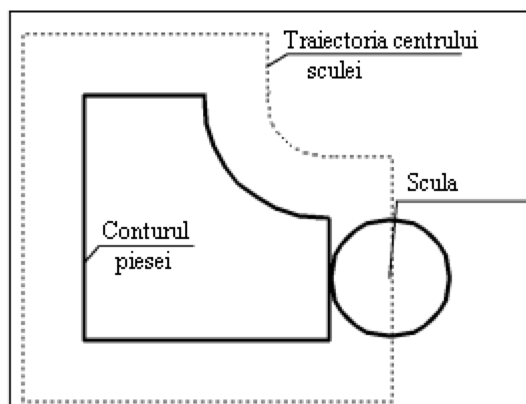


Fig. 1.2-9

Compensarea uzurii

În decursul prelucrării sculele se uzează. Se poate stabili o relație între uzură şi modificările dimensionale ale sculei (în lungime ca şi în rază). Se poate impune uzura sculei în comanda numerică. Pentru fiecare grup de compensare (la care se face referire la adresele H sau D) avem o valoare a geometriei, adică lungimea şi raza inițială a sculei, şi o valoare a uzurii. Când se impune compensarea, comanda numerică va compensa mişcarea cu suma celor două valori.

2 Axele controlate

Numărul de axe (în configurația de bază)	3 axe
În configurație extinsă	5 axe adiționale (8 axe în total)
Numărul de axe care pot fi mişcate simultan	8 axe (cu interpolare liniară)

2.1 Numele axelor

Numele axelor controlate poate fi definit în parametrii memoriei. Fiecare adresă poate fi atribuită uneia din axele fizice.
În configurația de bază, numele axelor pentru o comandă numerică de freză sunt X, Y şi Z.
Numele axelor adiționale (extindere) depind de tipurile lor.
Nume posibile pentru axele de extindere care execută deplasări liniare sunt: U, V şi W. Atunci când sunt paralele cu axele principale X, Y, şi Z, numele lor vor fi respectiv U, V si W.
Axele care execută mişcări de rotație sunt denumite A, B şi C. Axele de rotație a căror axă este paralelă la direcțiile X, Y şi Z sunt denumite respectiv A, B şi C.

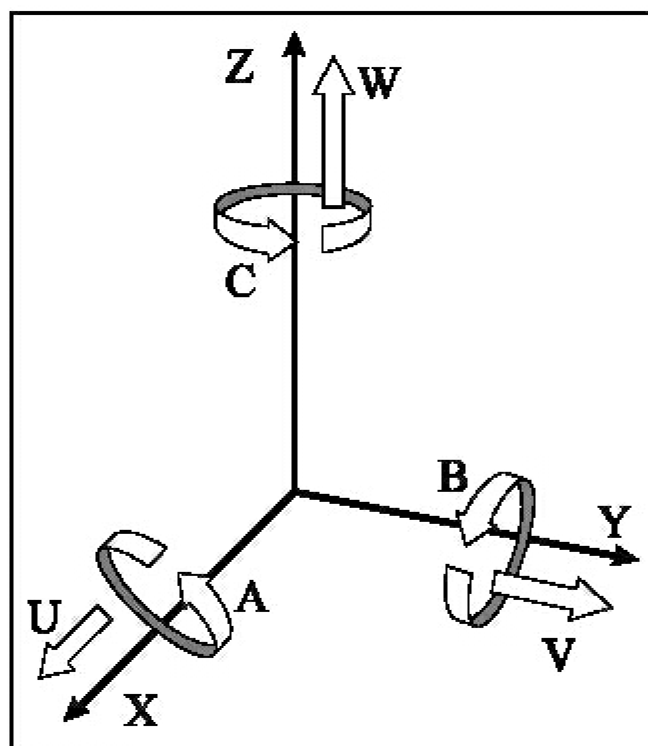


Fig. 2.1-1

2.2 Sistemul de unități şi incremente ale axelor

Datele coordonatelor pot fi specificate cu opt cifre. Ele pot avea şi semn. Semnul pozitiv + este omis.
Datele de introducere a lungimii pot fi specificate în mm sau inci. Acestea sunt unități de introducere pentru măsură. Unitatea dorită poate fi selectată prin program.
Sistemul de măsurare a traiectoriei al maşinii, poate măsura poziția în mm sau în inci. Acesta va determina unitatea de măsură de ieşire, care trebuie să fie specificată în comanda numerică ca un parametru. Cele două sisteme de unități de măsură nu pot fi combinate pe axele aceleeaşi maşini.
În cazul în care s-au impus unități de măsură diferite pentru datele de intrare şi de ieşire, comanda numerică va face conversia în mod automat.
Pentru axele de rotație, măsura este întotdeauna considerată în grade.

Sistemul de increment de intrare al comenzii numerice este considerat ca fiind cea mai mică unitate care poate fi introdusă. Aceasta poate fi selectată ca un parametru. Sunt disponibile trei sisteme de incremente IS-A, IS-B ŞI IS-C. Sistemele de incremente nu pot fi combinate pentru axele aceleeaşi maşini.
După ce a procesat datele de intrare, comanda numerică va furniza noua traiectorie pentru deplasarea axelor. Rezoluția lor este întotdeauna dublul incrementului sistemului de intrare ales. Acesta este denumit **sistemul de increment de ieşire** al comenzii numerice.
Practic sistemul de increment de intrare al comenzii numerice este determinat de către rezoluția codorului de poziție.

Sistemul de increment	Unitatea minimă care poate fi introdusă	Unitatea maximă care poate fi introdusă
IS-A	0.01 mm	999999.99 mm
	0.001 inci	99999.999 inci
	0.01 grade	999999.99 grade
IS-B	0.001 mm	99999.999 mm
	0.0001 inci	9999.9999 inci
	0.001 grade	99999.999 grade
IS-C	0.0001 mm	9999.9999 mm
	0.00001 inci	999.99999 inci
	0.0001 grade	9999.9999 grade

3 Funcții pregătitoare (coduri G)

Într-un bloc, tipul de comandă va fi determinat de adresa G şi de numărul care urmează. Tabelul următor conține codurile G interpretate de comanda numerică, grupele şi funcțiile pe care le conțin.

Cod G	Grupa	Funcția	Pagina
G00*	01	Poziționare	22
G01*		Interpolare liniară	22
G02		Interpolare circulară, elicoidală în sens orar (CW)	24
G03		Interpolare circulară, elicoidală în sens antiorar (CCW)	24
G04	00	Întârziere	51
G05.1		Modul multi bufer activ	
G07.1		Interpolare cilindrică	35
G09		Oprire precisă (în blocul dat)	48
G10		Impunere dată (programată)	58, 78
G11		Anulare impunere dată programată	
G12.1	26	Interpolarea în coordonate polare activată	31
G13.1		Interpolarea în coordonate polare dezactivată	31
G15*	17	Anulare comandă de coordonate polare	38
G16		Comandă de coordonate polare	38
G17*	02	Selectarea planului X_pY_p	61
G18*		Selectarea planului Z_pX_p	61
G19		Selectarea planului Y_pZ_p	61
G20	06	Intrare în inci	40
G21		Intrare în metric	40
G22*	04	Activarea funcției de verificare programabilă a cursei	157
G23		Dezactivarea funcției de verificare programabilă a cursei	157
G25*	8	Dezactivarea detecției fluctuației vitezei arborelui	66
G26		Activarea detecției fluctuației vitezei arborelui	66
G28	0	Întoarcere programată în punctul de referință	52
G29		Revenire din punctul de referință	53
G30		Revenire la primul, al doilea, al treilea şi al patrulea punct de referință	53
G31		Funcția de salt	154
G33	01	Filetare	29
G37	00	Măsurarea automată a lungimii sculei pe Z	155
G38		Reținerea vectorului de compensare a sculei	99
G39		Compensarea sculei la arcele de colț	99

Cod G	Grupa	Funcția	Pagina
G40*	07	Anularea compensării razei / tridimensionale a sculei	84
G41		Compensarea razei sculei la stânga / compensarea tridimensională a sculei	84, 87
G42		Compensarea razei sculei la dreapta	84, 87
G43*	08	Compensarea lungimii sculei +	79
G44*		Compensarea lungimii sculei -	79
G45		Creşterea ofsetului sculei	80
G46		Descreşterea ofsetului sculei	80
G47		Creşterea dublă a ofsetului sculei	80
G48		Descreşterea dublă a ofsetului sculei	80
G49*	08	Anularea compensării lungimii sculei	79
G50*	11	Anulare modificare la scară	115
G51		Modificare la scară	115
G50.1*	18	Anulare imagine în oglindă programabilă	116
G51.1		Imagine în oglindă programabilă	116
G52	00	Impunere sistem de coordonate local	59
G53		Poziționare în sistemul de coordonate al maşinii	56
G54*	14	Selectare sistem de coordonate de lucru 1	57
G55		Selectare sistem de coordonate de lucru 2	57
G56		Selectare sistem de coordonate de lucru 3	57
G57		Selectare sistem de coordonate de lucru 4	57
G58		Selectare sistem de coordonate de lucru 5	57
G59		Selectare sistem de coordonate de lucru 6	57
G61	15	Modul tăiere precisă	48
G62		Modul corecție automată la colț	49
G63		Inhibare corecții	49
G64*		Tăiere continuă	49
G65		Apel macro simplu	160
G66		Apel macro modal (A) în fiecare comandă de mişcare	161
G66.1		Apel macro modal (B) din fiecare bloc	162
G67		Anulare apel macro modal (A/B)	161
G68	16	Rotire sistem de coordonate	114
G69*		Anulare rotire sistem de coordonate	114
G73	01	Ciclu de găurire cu viteză mare şi retrageri pentru ruperea şpanului	137
G74		Ciclu de filetare în sens antiorar	139
G76		Ciclu de alezare fină	140
G80*		Anulare ciclu memorat	141

Cod G	Grupa	Funcția	Pagina
G81		Găurire, ciclu de alezare parțială	141
G82		Găurire, ciclu de alezare la retragere	142
G83		Ciclu de găurire cu retrageri pentru ruperea șpanului	143
G84		Ciclu de filetare cu tarod	144
G84.2		Ciclu de filetare rigidă cu tarod	145
G84.3		Ciclu de filetare rigidă cu tarod în sens antiorar	145
G85		Ciclu de alezare	148
G86		Ciclu de alezare cu retragerea sculei cu avans rapid	149
G87		Ciclu de alezare /ciclu de alezare la retragere	150
G88		Ciclu de alezare (operare manuală în punctul de fund)	152
G89		Ciclu de alezare (întârziere în punctul de fund, retragere cu avans de lucru)	153
G90*	03	Comandă absolută	38
G91*		Comandă incrementală	38
G92	00	Schimbare coordonate de lucru \ impunere turație maximă arbore	58
G94*	05	Avans pe minut	45
G95*		Avans pe rotație	45
G96	13	Controlul vitezei periferice constante	64
G97*		Anularea controlului vitezei periferice constante	64
G98*	10	Revenirea sculei la nivelul inițial după execuția ciclului memorat	133
G99		Revenirea sculei la nivelul punctului R după execuția ciclului memorat	133

Note:

- Semnul * care marchează codurile G dintr-un grup, reprezintă starea în care trece comanda numerică la punerea sub tensiune.
- Dacă într-un grup sunt marcate cu * mai multe coduri, se poate impune printr-un parametru codul care va fi activat la punerea sub tensiune. Acestea sunt: G00, G01; G17, G18; G43, G44, G49; G90, G91; G94, G95.
- La punerea sub tensiune, dintre G20 şi G21 va fi activă cea care era impusă în momentul decuplării de la rețea.
- Interpretarea implicită a comenzii G05.1 după punerea sub tensiune poate fi specificată cu parametrul MULBUF.
- Codurile G din grupul 00 nu sunt modale; restul codurilor sunt.
- Într-un bloc se pot scrie mai multe coduri G cu restricția ca să fie folosit doar unul din grupul cu aceeaşi funcție.
- Referirea la un cod G ilegal sau specificarea mai multor coduri G care aparțin aceluiaşi grup în acelaşi bloc va produce un mesaj de eroare „ 3005 ILLEGAL G CODE “ (*cod G incorect*).

4 Interpolarea

4.1 Poziționarea (G00)

Seria de instrucțiuni

G00 v

se referă la poziționarea în sistemul de coordonate curent.
Deplasarea se face la coordonata v. Indicația v (vector) se referă aici (şi pe mai departe) la toate axele controlate folosite pe maşina-unealtă. (Ele pot fi X, Y, Z, U, V, W, A, B, C).
Poziționarea se realizează de-a lungul unei linii drepte implicând mişcările simultane ale tuturor axelor specificate în bloc. Coordonatele pot fi date absolute sau incrementale.
Viteza de poziționare nu poate fi comandată în program deoarece se realizează cu valori diferite pentru fiecare axă, valori impuse de constructorul maşinii ca parametri. Atunci când sunt mişcate simultan mai multe axe, este calculată de către comanda numerică rezultanta vectorială a vitezei astfel încât să se execute poziționarea în intervalul minim de timp, iar viteza să nu depăşească nicăieri parametrul de avans rapid impus pentru fiecare axă.

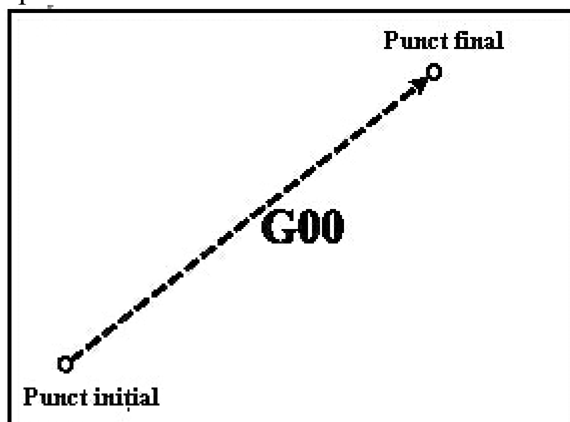


Fig. 4.1-1

În execuția instrucțiunii G00, comanda numerică execută accelerări şi frânări la începutul şi respectiv sfârşitul mişcărilor. La terminarea mişcării, comanda numerică va verifica semnalul „poziție atinsă" atunci când parametrul *POSCHECK* din câmpul parametrilor este1, sau nu-l va verifica dacă parametrul este pus pe 0. Se va aştepta apariția semnalului „poziție atinsă" timp de 5 secunde şi, dacă nu ajunge acest semnal se va afişa mesajul *1020 POSITION ERROR (eroare de poziție).* Deviația maximă acceptabilă de la poziție poate fi specificată în parametrul *INPOS*.
Fiind un cod modal, G00 rămâne activ până când este reînscris de către o altă comandă de interpolare. După punerea sub tensiune, este activ G00 sau G01, funcție de valoarea impusă în grupul de parametri *CODES*.

4.2 Interpolarea liniară (G01)

Seria de instrucțiuni

G01 v F

va selecta un mod de interpolare liniară. Data scrisă pentru v poate fi valoare absolută sau incrementală, interpretată în sistemul de coordonate curent. Viteza de deplasare (avansul) poate fi programat la adresa F.
Avansul programat la adresa F va fi realizat invariabil de-a lungul traiectoriei programate. Componentele sale axiale sunt:

Avansul pe axa X este $F_x = \frac{X}{L} F$

Avansul pe axa Y este $F_Y = \frac{Y}{L} F$

……………………..

Avansul pe axa U este $F_u = \frac{u}{L} F$

……………………

Avansul pe axa C este $F_c = \frac{c}{L} F$

unde x, y, u, c sunt deplasările programate de-a lungul axelor respective, L este lungimea vectorială a deplasării:

$$L=\sqrt{x^2+y^2+...+u^2+...+c^2}$$

```
G01 X192 Y80 F150
```

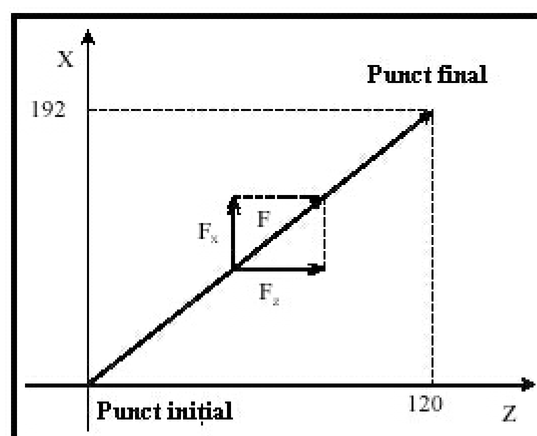


Fig. 4.2-1

Avansul de-a lungul unei axe de rotație este interpretat în unități de grad pe minut (°/min):

```
G01 B270 F120
```

În blocul de mai sus, F120 semnifică 120 grade /minut.
Dacă sunt combinate prin interpolare liniară o mişcare rectilinie cu o mişcare de rotație, componentele avansului vor fi distribuite în acord cu formula de mai sus. De exemplu în blocul

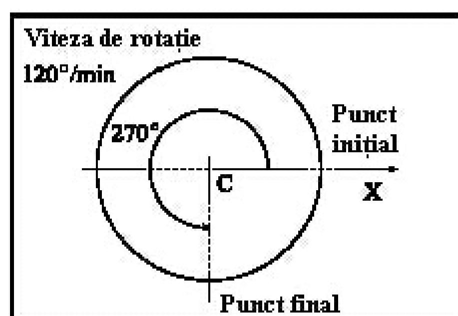


Fig. 4.2-2

```
G91 G01 Z100 B45 F120
```

componentele avansului în direcțiile Z şi B sunt:

avansul pe Z $F_z = \frac{100}{\sqrt{100^2+45^2}} 120 = 109.4$ mm/min

avansul pe B $F_b = \frac{45}{\sqrt{100^2+45^2}} 120 = 49.2$ °/min

Fiind un cod modal, G01 este activ până când se înscrie cu o altă comandă de interpolare. La punerea sub tensiune, este activ fie G00 fie G01, funcție de valoarea impusă parametrilor în grupul *CODES* din câmpul parametrilor.

4.3 Interpolarea circulară şi spirală

$$G17 \begin{Bmatrix} G02 \\ \hline G03 \end{Bmatrix} X_p \; Y_p \begin{Bmatrix} R \\ \hline I \; J \end{Bmatrix} F$$

$$G18 \begin{Bmatrix} G02 \\ \hline G03 \end{Bmatrix} X_p \; Z_p \begin{Bmatrix} R \\ \hline I \; K \end{Bmatrix} F$$

$$G19 \begin{Bmatrix} G02 \\ \hline G03 \end{Bmatrix} Y_p \; Z_p \begin{Bmatrix} R \\ \hline J \; K \end{Bmatrix} F$$

Aceste serii de instrucțiuni specifică interpolarea circulară.
O interpolare circulară este realizată în planul selectat prin comenzile G17, G18, G19 în sens orar sau antiorar (cu G02 respectiv G03).

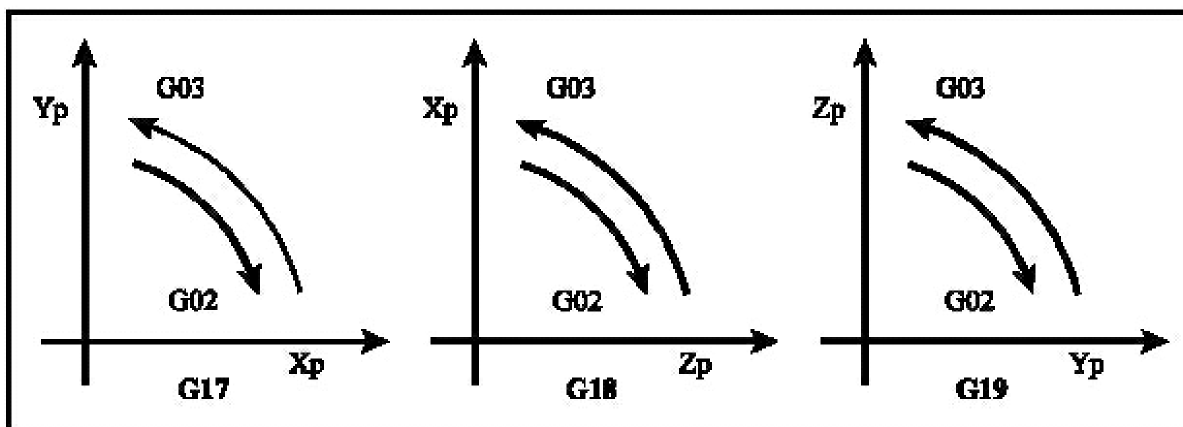


Fig. 4.3-1

Aici şi pe mai departe, semnificațiile pentru X_p, Y_p, şi Z_p sunt:

X_p : axa X sau o axă paralelă la ea,
Y_p : axa Y sau o axă paralelă la ea,
Z_p : axa Z sau o axă paralelă la ea.

Valorile pentru X_p, Y_p, şi Z_p sunt coordonatele punctului final al cercului în sistemul de coordonate dat, specificate ca date absolute sau incrementale.

Alte date pentru cerc pot fi specificate într-unul din următoarele două moduri.

Cazul 1

La adresa R, unde R este raza cercului. Acum comanda numerică va calcula automat coordonatele centrului cercului din coordonatele punctului de start (punctul în care se află comanda numerică în momentul în care s-a introdus blocul pentru cerc), coordonatele punctului de sfârşit (valorile definite la adresele X_p, Y_p, şi Z_p) şi din raza cercului programat, R. Deoarece se pot desena două cercuri diferite cu raza R între punctele de început şi cel de sfârşit pentru o direcție de parcurgere dată (G02 sau G03), comanda numerică va interpola un arc mai mic sau mai mare de 180° când raza cercului este specificată ca număr pozitiv sau respectiv număr negativ. De exemplu:

Secțiunea de arc 1: `G02 X50 Y40 R40`
Secțiunea de arc 2: `G02 X50 Y40 R-40`
Secțiunea de arc 3: `G03 X50 Z40 R40`
Secțiunea de arc 4: `G03 X50 Z40 R-40`

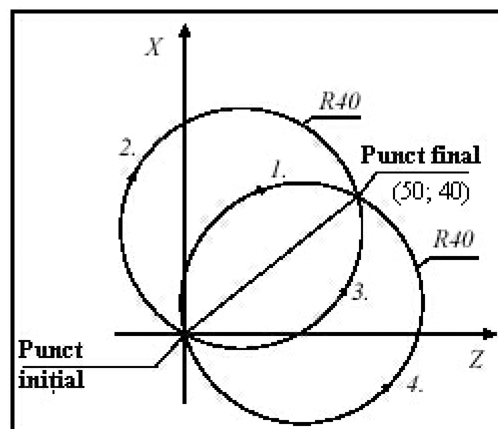


Fig. 4.3-2

Cazul2

Centrul cercului este specificat la adresele I, J, K pentru axele X_p, Y_p, şi Z_p. Valorile specificate la adresele I, J, K sunt interpretate întotdeauna ca incrementale de către comanda numerică, astfel că vectorul definit cu valorile punctelor I, J, K este îndreptat din punctul de început către centrul cercului. De exemplu:

Cu G17: `G03 X10 Y70 I-50 J-20`
Cu G18: `G03 X70 Z10 I-20 K-50`
Cu G19: `G03 Y10 Z70 J-50 K-20`

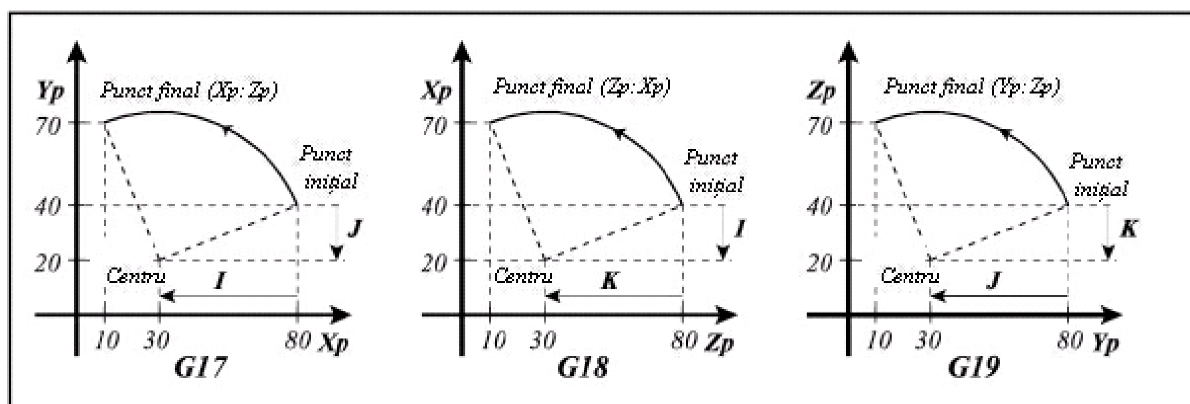


Fig. 4.3-3

Avansul pe traiectorie poate fi programat la adresa F, fiind îndreptat în direcția tangentei la cerc şi menținându-se constant la parcurgerea traiectoriei.

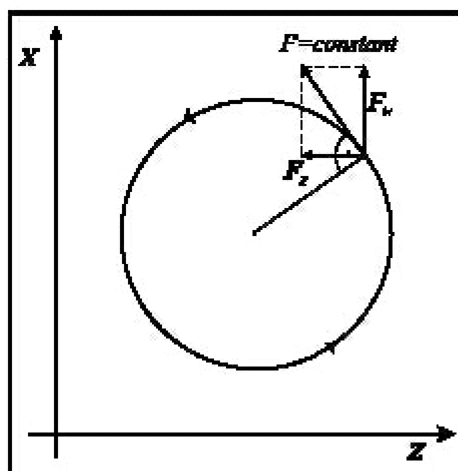


Fig. 4.3-4

Note:

- I0, J0, K0 pot fi omise:

  `G03 X0 Y100 I-100`

- Când X_p, Y_p, şi Z_p sunt omise toate, sau coordonatele punctului final coincid cu cele ale punctului de start, atunci:

  a. dacă coordonatele centrului cercului sunt programate la adresa I, J, K, comanda numerică va interpola un cerc complet de 360° . Exemplu : `G03 I-100`

  b. dacă este programată raza R, comanda numerică afişează eroarea *3012 ERRONEOUS CIRCLE DEF. R (raza cercului R definită eronat).*

- Când blocul cercului fie

  a. nu conține nici raza (R) nici I, J, K ,

  b. face referire la adresa I, J, K în afara planului selectat, comanda numerică afişează eroarea *3014 ERRONEOUS CIRCLE DEF. (cercul definit eronat).*

  Exemple: `G03 X0 Y100`, sau `(G18) G02 X0 Z100 J-100`.

- Comanda numerică afişează mesajul *3011 RADIUS DIFFERENCE (diferență de rază)* de câte ori diferența pe rază între punctul de pornire şi punctul final al cercului definit în blocurile G02, G03 depăşeşte valoarea definită în parametrul *RADDIF*.

De câte ori diferența de rază este mai mică decât valoarea specificată în parametrul de mai sus, comanda numerică va deplasa scula de-a lungul unei traiectorii spirale în care raza variază liniar funcție de unghiul central.

Viteza unghiulară, şi nu cea tangențială la traiectorie va fi constantă la interpolarea unui arc de cerc cu rază variabilă.

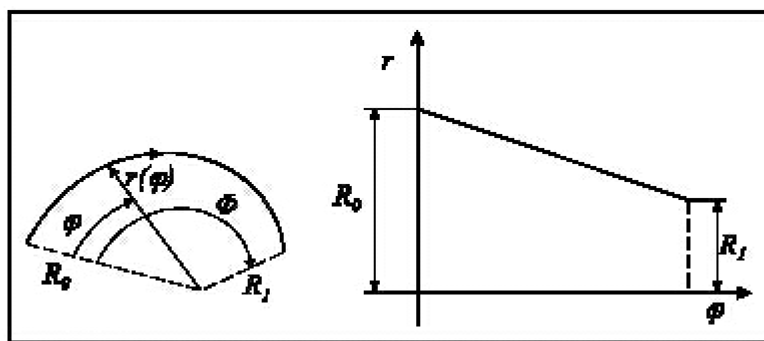


Fig. 4.3-5

Următorul detaliu de program este un exemplu de cum se poate specifica o interpolare spirală (cerc cu rază variabilă) prin folosirea adreselor I, K:

```
G17 G90 G0 X50 Y0
G3 X-20 I-50
```

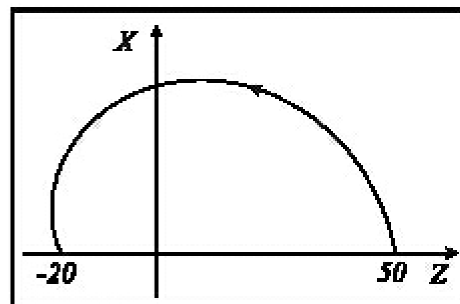


Fig. 4.3-6

Dacă raza specificată a cercului este mai mică decât jumătate din distanța liniei drepte care interconectează punctul de plecare cu cel final, comanda numerică va considera raza specificată a cercului ca rază-punct de plecare şi va interpola un cerc cu rază variabilă (spirală), al cărui punct de plecare este localizat pe linia dreaptă care uneşte punctul de plecare cu cel final, la distanța R de punctul de plecare.

```
G17 G0 G90 X0 Y0
G2 X40 Y30 R10
```

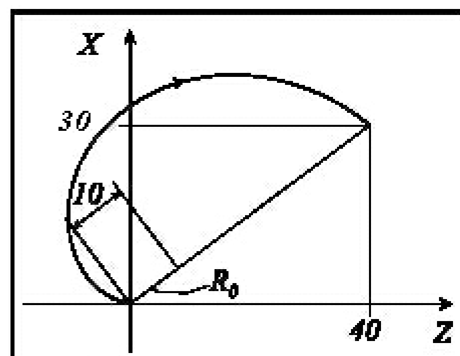


Fig. 4.3- 7

4.4 Interpolarea elicoidală (G03, G02)

$$G17 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X_p \; Y_p \; q \begin{Bmatrix} R \\ I \; J \end{Bmatrix} F$$

$$G18 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X_p \; Z_p \; q \begin{Bmatrix} R \\ I \; K \end{Bmatrix} F$$

$$G19 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} Y_p \; Z_p \; q \begin{Bmatrix} R \\ J \; K \end{Bmatrix} F$$

Aceste serii de instrucțiuni vor defini o interpolare elicoidală.
Deosebirea față de interpolarea circulară este cea de a treia axă (q) care nu este o axă componentă a planului cercului. Comanda numerică va executa doar o deplasare simplă de-a lungul axei q.

Avansul specificat la adresa F este activ doar pentru traiectoria circulară. Componenta F_q de-a lungul axei q este obținută din relația:

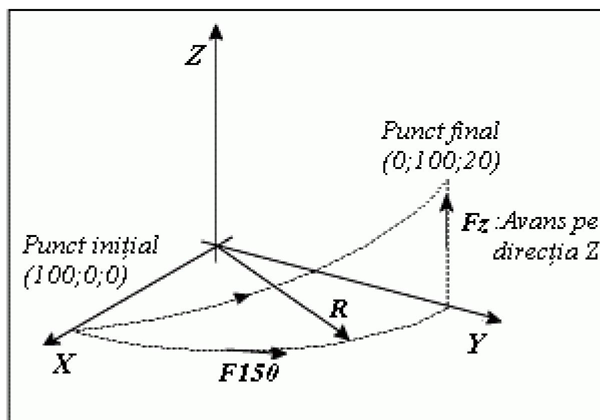


Fig. 4.4-1

$$F_q = \frac{L_q}{L_{arc}} F$$

Unde:

L_q: deplasare de-a lungul axei q,
L_{arc}: lungimea arcului circular,
F: avansul programat,
F_q: avansul de-a lungul axei q.

De exemplu:

```
G17 G03 X0 Y100 Z20 R100 F150
```

Seria de instrucțiuni

$$G17 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X_p\ Y_p\ q\ r\ s \begin{Bmatrix} R \\ I\ J \end{Bmatrix} F$$

$$G18 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X_p\ Z_p\ q\ r\ s \begin{Bmatrix} R \\ I\ K \end{Bmatrix} F$$

$$G19 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} Y_p\ Z_p\ q\ r\ s \begin{Bmatrix} R \\ J\ K \end{Bmatrix} F$$

defineşte o interpolare spațială multidimensională în care q, r, s sunt axe opționale care nu sunt implicate în interpolarea circulară.
De exemplu, seria de instrucțiuni

```
G17 G3 X0 Y-100 Z50 V20 I-100
```

va deplasa scula pe suprafața unui cilindru oblic, dacă V este o axă paralelă la Y.

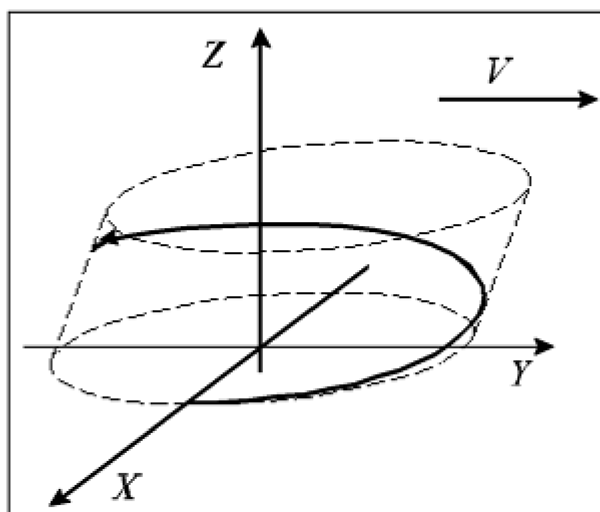


Fig. 4.4-2

Note:

- Atunci când parametrul HELICALF din câmpul parametrilor este impus 1, comanda numerică va executa avansul programat de-a lungul traiectoriei spațiale.
- În cazul în care cercul specificat în planul selectat are raza variabilă, interpolarea se va executa de-a lungul suprafeței conului specificat.

- Compensarea specificată pentru raza sculei se va face invariabil în planul cercului.

4.5 Filetarea cu pas constant (G33)

Instrucțiunile

G33 v F Q

G33 v E Q

vor defini o filetare cu pas constant cilindrică sau conică.

Pentru vectorul v pot fi scrise coordonatele a maximum două axe. Comanda numerică va executa un filet conic dacă sunt impuse două coordonate vectorului v. Comanda numerică va considera că spira se înfăşoară de-a lungul axei lungi.

Dacă α<45°, adică Z>X, se va lua în considerare că spira se înfăşoară de-a lungul axei Z.

Dacă α>45°, adică X>Z, se va lua în considerare că spira se înfăşoară de-a lungul axei X.

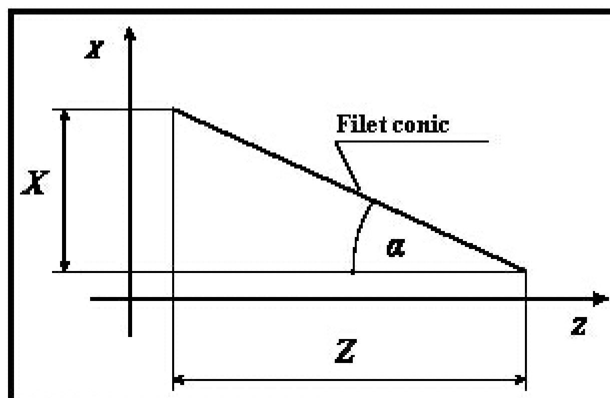


Fig. 4.5-1

Pasul filetului poate fi definit în unul din următoarele două moduri:

- dacă pasul este specificat la adresa F, data va fi interpretată ca mm/rot sau inci/rot. Astfel, dacă trebuie tăiat un filet cu pas de 2,5 mm trebuie programat F2.5 .
- dacă pasul este specificat la adresa E, comanda numerică va executa un filet în inci. Adresa E este interpretată ca număr de spire pe inci. De exemplu, dacă e programat E3, se va executa un filet cu pas 1/3”=25.4/3=8.4667 mm.

Unghiul începutului filetului este specificat la adresa Q, în grade față de pulsul de zero al codorului arborelui principal. Printr-o programare adecvată a valorii pentru Q, se poate executa şi un filet cu mai multe începuturi, adică se poate programa comenzii numerice o anumită dispunere a arborelui la care trebuie tăiat un anumit început al filetului. Dacă de exemplu trebuie tăiat un filet cu două începuturi, primul început va fi la Q0 iar al doilea la Q180 (fără nici o altă programare specială).

G33 este o funcție modală. Dacă sunt programate câteva blocuri de filetare în succesiune, filetele pot fi executate în orice suprafață arbitrară limitată de linii drepte.



Fig. 4.5-2

Comanda numerică este sincronizată la pulsul de zero al codorului arborelui principal în primul bloc, aşa că nu se va mai executa nici o sincronizare în blocurile consecutive şi va rezulta un filet continuu în fiecare secțiune a liniilor. Tot în primul bloc se ia în considerație şi unghiul începutului filetului (Q) programat.

Un exemplu de programare a filetării:

```
N50 G90 G0 X0 Y0 S100 M4
N55 Z2
N60 G33 Z-100 F2
N65 M19
N70 G0 X5
N75 Z2 M0
N80 X0 M4
N85 G4 P2
N90 G33 Z-100 F2
...
```

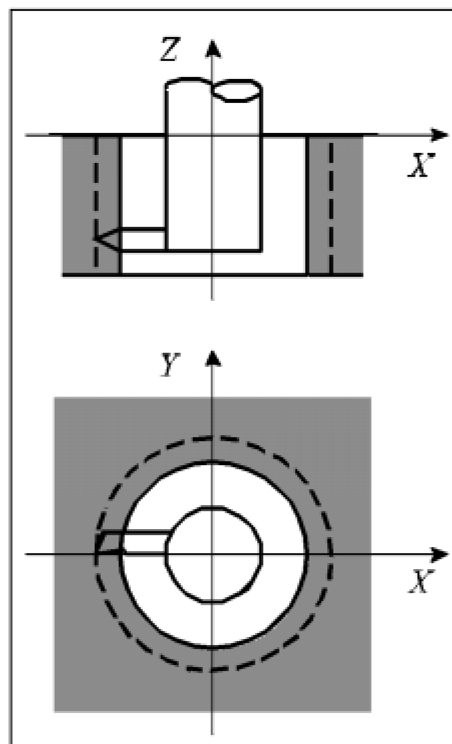


Fig. 4.5-3

Explicații:

N50, N55 – Deplasarea sculei în centrul găurii, pornirea rotirii arborelui în sens antiorar,
N60 – Primul ciclu de filetare, (pas 2mm),
N65 – Oprire orientată a arborelui (arborele este oprit într-o poziție determinată)
N70 – Retragerea sculei pe axa X,
N75 – Retragerea sculei la începutul găurii, oprire programată, operatorul ajustează scula pentru următorul ciclu de filetare,
N80 – Revenire în centrul găurii, repornire arbore,
N85 – Aşteptarea atingerii turației de către arbore,
N90 – Al doilea ciclu de filetare.

Note:

- Se va afişa mesajul de eroare *3020 DATA DEFINITION ERROR G33 (eroare la definirea datei pentru G33)* dacă au fost specificate mai mult de două coordonate simultan în blocul de filetare, sau sunt specificate simultan ambele adrese F şi E.
- Se va afişa mesajul de eroare *3022 DIVIDE BY 0 IN G33 (divizare prin 0 în G33)* dacă s-a specificat 0 pentru adresa E în blocul de filetare.
- Pentru a putea executa comanda G33 trebuie ca pe arbore să fie montat un codor de poziție.
- Pe parcursul execuției comenzii G33, comanda numerică va considera în mod automat valorile corecțiilor pentru avansuri şi rotire arbore ca fiind 100%; de asemenea tasta stop va avea efect doar după ce se execută blocul în curs.
- Cu scopul de a obține un pas constant al filetului pe toată piesa prelucrată, trebuie asigurată cursă liberă pentru sculă la ambele capete ale piesei de prelucrat.
- În cursul filetării avansul (în mm/min) nu poate depăşi valoarea selectată în grupa de parametri FEEDMAX.
- În cursul filetării viteza arborelui (în rot/min) nu poate depăşi valoarea maximă admisă pentru codorul de poziție, atât din punct de vedere mecanic cât şi electric (frecvența maximă la ieşire).

4.6 Interpolarea în coordonate polare (G12.1, G13.1)

Interpolarea în coordonate polare este o metodă de operare a comenzii numerice, în cazul în care descrierea piesei în sistemul de coordonate carteziene are traiectoria conturului realizată prin deplasarea unei axe liniare şi a unei axe de rotaţie.
Comanda

G12.1

activează modul de interpolare în coordonate polare. Traiectoria sculei de frezare poate fi descrisă în programul piesă care urmează într-un sistem de coordonate carteziene în modul obişnuit prin programarea interpolărilor liniare şi circulare, ţinând cont de compensarea razei sculei. *Această comandă trebuie să fie dată într-un bloc separat şi după ea nu se poate scrie o altă comandă.*
Comanda

G13.1

dezactivează modul de interpolare în coordonate polare. *Această comandă trebuie să fie dată într-un bloc separat şi după ea nu se poate scrie o altă comandă.* La punerea sub tensiune sau după reset se trece automat în starea G13.1.

Selectarea planului

Trebuie selectat un plan care să determine adresele axei liniare şi a celei de rotaţie înainte de a activa interpolarea în coordonate polare.

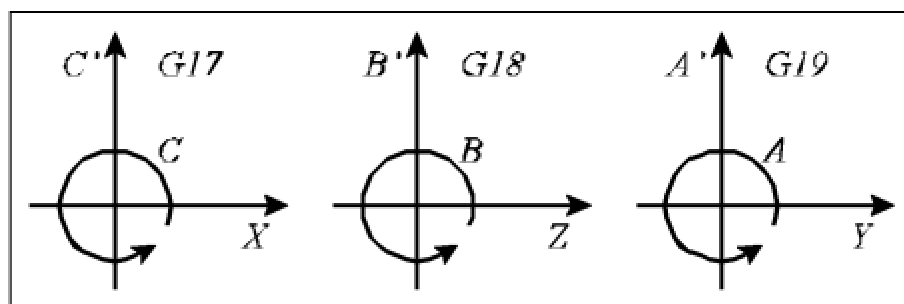


Fig. 4.6-1

Comanda

G17 X_ C_

selectează axa X ca axă liniară şi axa C ca axă de rotaţie. În desen este indicată cu C' axa virtuală, a cărei programare este executată prin definirea măsurilor lungimii.
Prin intermediul comenzilor

G18 Z_ C_
G19 Y_ C_

se pot selecta alte perechi de axe liniare şi de rotaţie în modul prezentat mai sus.

Ofsetul punctului de zero al piesei în decursul interpolării în coordonate polare

În cazul folosirii interpolării în coordonate polare originea sistemului de coordonate al piesei trebuie aleasă astfel încât să coincidă cu axa de rotaţie a axei circulare.

Poziţia axelor în momentul activării interpolării în coordonate polare

Înainte de activarea interpolării în coordonate polare (comanda G12.1) asiguraţi-vă că ***poziţia***

axei circulare este la 0. ***Poziția axei liniare*** poate fi pozitivă sau negativă dar ***nu poate fi pe 0.***

Programarea coordonatelor lungimii în cursul interpolării în coordonate polare
În starea activă a interpolării în coordonate polare, data coordonatei lungimii poate fi programată pe ambele axe care aparțin planului selectat. Axa de rotație în planul selectat funcționează ca a doua axă (virtuală). Dacă de exemplu axele X şi C au fost selectate cu ajutorul comenzii G17 X_C_ , adresa C poate fi programată ca şi axa Y în cazul selectării planului G17 X_Y_.
Programarea primei axe fiind ca diametru, nu influențează programarea ***axei virtuale***, data de coordonată trebuind să fie introdusă întotdeauna ca ***rază*** pentru ***axa virtuală***. Dacă de exemplu este executată interpolare în coordonate polare în planul X C, valoarea scrisă la adresa C trebuie specificată ca rază, independent dacă adresa X este dată ca diametru sau ca rază.

Deplasarea pe axele care nu iau parte la interpolarea în coordonate polare
Deplasarea pe aceste axe se desfăşoară normal, independent de starea activată a interpolării în coordonate polare.

Programarea interpolării circulare în cursul interpolării în coordonate polare
Definirea unui cerc în modul de interpolare în coordonate polare este posibilă după cum se ştie cu ajutorul razei sau prin programarea coordonatelor centrului cercului. În cel de al doilea caz adresele I, J, K trebuie să fie folosite în acord cu planul selectat după cum se vede mai jos:

```
G17 X_ C_              G18 Z_ B_              G19 Y_ A_
G12.1                  G12.1                  G12.1
...                    ...                    ...
G2 (G3) X_ C_ I_ J_    G2 (G3) B_ Z_ I_ K_    G2 (G3) Y_ A_ J_ K_
```

Folosirea compensării razei sculei în cursul interpolării în coordonate polare
Comenzile G41, G42 se pot folosi în modul obişnuit în cursul interpolării în coordonate polare. Se vor avea în vedere următoarele restricții:
- Activarea interpolării în coordonate polare (comanda G12.1) este posibilă numai în starea G40,
- Dacă este activat G41 sau G42 în starea G12.1, trebuie programat G40 înainte de a se dezactiva interpolarea în coordonate polare (comanda G13.1).

Restricții privind programarea în cursul interpolării în coordonate polare
Comenzile de mai jos nu pot fi utilizate în starea activată a interpolării în coordonate polare:
- schimbarea planului: G17, G18, G19,
- transformările de coordonate: G52, G92,
- schimbarea sistemului de coordonate de lucru: G54, …, G59,
- orientarea în sistemul de coordonate al maşinii: G53.

Avansul în cursul interpolării în coordonate polare
Interpretarea avansului în interpolarea în coordonate polare este viteza tangențială ca în cazul interpolării în unghi drept: este definită viteza relativă a piesei şi sculei.
Traiectoria descrisă într-un sistem de coordonate cartezian este executată la interpolarea în coordonate polare prin deplasarea unei axe liniare şi a unei axe de rotație. Pe măsură ce

centrul sculei se apropie de axa circulară de rotație, axa de rotație ar trebui să execute paşi din ce în ce mai mari în aceeaşi unitate de timp pentru că viteza pe traiectorie este constantă. Totuşi viteza maximă permisă pentru axa de rotație este definită prin parametrul care limitează viteza axei circulare. De aceea, în apropierea originii comanda numerică reduce viteza pas cu pas pentru a nu se depăşi limita de viteză pe axa de rotație.

Figura alăturată prezintă cazurile în care sunt programate linii drepte paralele cu axa X (1, 2, 3, 4). Deplasarea Δx se face cu avansul programat într-o unitate de timp. Pentru fiecare linie dreaptă (1, 2, 3, 4) corespunde câte o deplasare unghiulară (φ_1, φ_2, φ_3, φ_4) pentru deplasarea Δx. Aparent, cu cât prelucrarea se apropie de origine, cu atât mai mare este deplasarea unghiulară pe care trebuie să o execute axa unghiulară în unitatea de timp pentru a menține avansul programat.

În cazul în care deplasarea unghiulară de executat în unitatea de timp depăşeşte valoarea parametrului FEEDMAX impus pentru axa de rotație, comanda numerică va reduce treptat avansul tangențial.

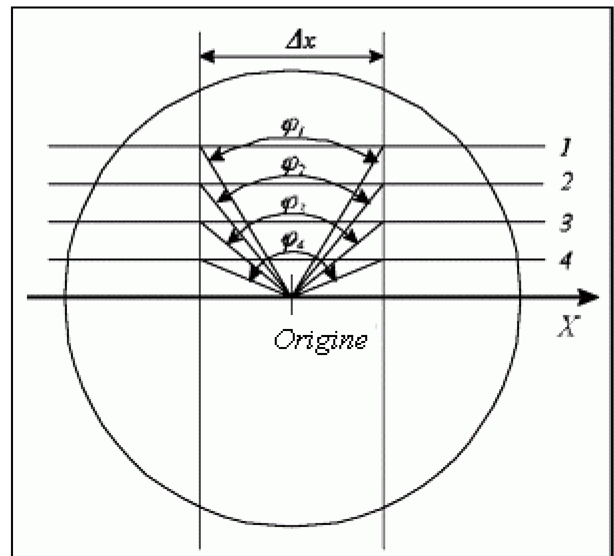


Fig. 4.6-2

Ținând cont de aceasta,trebuie evitate programele în care centrul sculei se apropie de origine.

Exemplu

În figură se prezintă un exemplu pentru utilizarea interpolării în coordonate polare. Axele care iau parte la interpolare sunt: axa X (axa liniară) şi axa C (axa de rotație). Axa X este programată în diametru, în timp ce axa C este programată în rază.

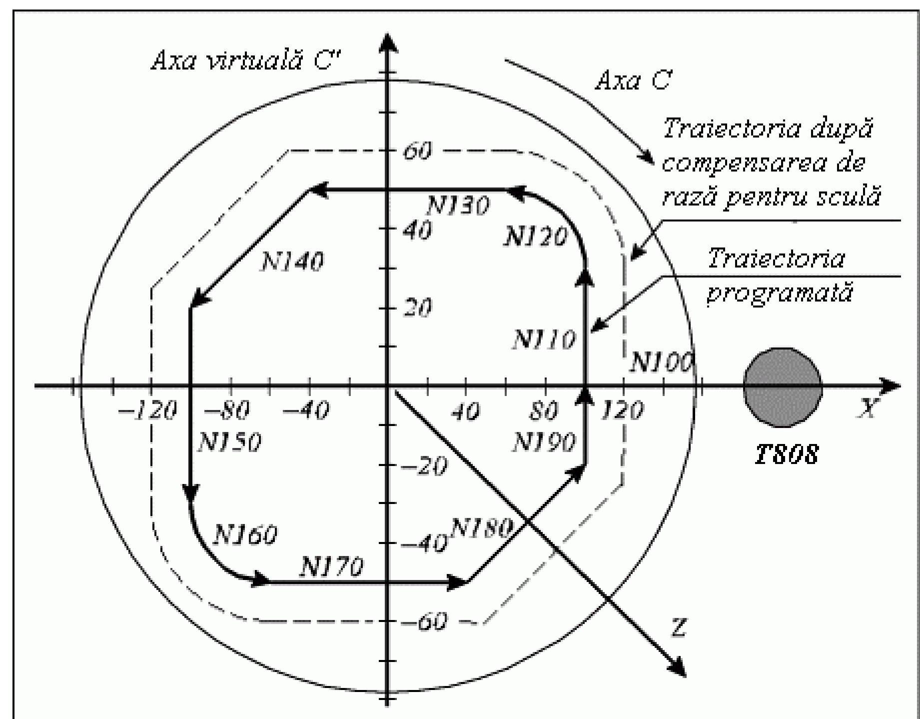


Fig. 4.6-3

4.6 Interpolarea în coordonate polare (G12.1, G13.1)

```
%O7500(INTERPOLARE ÎN COORDONATE POLARE)
...
N050 T808
N060 G59              (poziția de start a sistemului de coordonate
G59
                       în direcția X pe axa de rotație C)
N070 G17 G0 X200 C0   (selectare plan X, C; orientare la
                       coordonatele X diferit de 0, C=0)
N080 G94 Z-3 S1000 M3
N090 G12.1            (activare interpolare în coordonate polare)
N100 G42 G1 X100 F1000
N110 C30
N120 G3 X60 C50 I-20 J0
N130 G1 X-40
N140 X-100 C20
N150 C-30
N160 G3 X-60 C-50 R20
N170 G1 X40
N180 X100 C-20
N190 C0
N200 G40 G0 X150
N210 G13.1                (dezactivare interpolare în coordonate
polare)
N220 G0 G18 Z100          (Retragere sculă, selectare plan X, Z)
...
%
```

4.7 Interpolarea cilindrică (G7.1)

Atunci când trebuie frezat un canal de camă cilindrică pe suprafaţa unui cilindru, trebuie utilizată interpolarea cilindrică. În acest caz axa de rotaţie a cilindrului şi axa de rotaţie trebuie să coincidă.
Mişcările axei de rotaţie sunt specificate în program în grade, care sunt convertite în mişcări liniare de-a lungul suprafeţei de către comanda numerică în funcţie de raza cilindrului, astfel încât interpolările liniară şi circulară pot fi programate împreună cu o altă axă liniară.
Mişcările rezultate după interpolare sunt reconvertite în mişcări în grade pentru axa de rotaţie.
Activarea interpolării cilindrice se face cu comanda

G7.1 Qr

unde:

Q: adresa axei de rotaţie care ia parte la interpolarea cilindrică

r: raza cilindrului

Dacă de exemplu axa de rotaţie folosită în interpolarea cilindrică este axa C şi raza cilindrului este 50 mm, interpolarea cilindrică este activată cu ajutorul comenzii G7.1 C50.
În programul piesă care urmează traiectoria care urmează să fie frezată pe suprafaţa cilindrică poate fi descrisă prin specificarea interpolării liniare şi circulare. Coordonata pentru axa liniară trebuie să fie dată în mm, în timp ce cea pentru axa de rotaţie în grade.
Dezactivarea interpolării cilindrice se face cu comanda

G7.1 Q0

deşi codul corespunde cu cel pentru activare, singura diferenţă fiind adresa axei de rotaţie care este 0.
Interpolarea cilindrică indicată în exemplul de mai sus (G7.1 C50) se poate dezactiva prin intermediul comenzii G7.1 C0.
Comanda G7.1 trebuie să fie dată într-un bloc separat.

<u>Selectarea planului</u>

Codul pentru selectarea planului este determinat întotdeauna de către numele axei liniare paralele cu axa de rotaţie. Axele de rotaţie paralele cu axele X, Y, şi Z sunt axele A, B şi respectiv C.

G17 X A sau	G18 Z C sau	G19 Y B sau
G17 B Y	G18 A X	G19 C Z

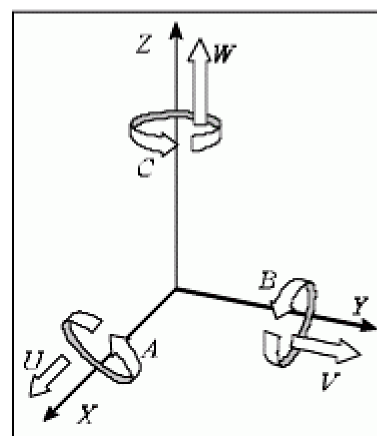


Fig. 4.7-1

<u>Interpolarea circulară</u>

Este posibil să se definească interpolarea circulară în modul de interpolare cilindrică, însă doar prin specificarea razei R.
Nu se poate executa nici o interpolare circulară în cazul interpolării cilindrice dacă se introduce centrul cercului (I, J, K).
Raza cercului este interpretată întotdeauna în mm sau inci, niciodată în grade.
De exemplu interpolarea circulară dintre axele Z şi C poate fi specificată în două moduri:

G18 Z_ C_	G19 C_ Z_
G2 (G3) Z_ C_ R_	G2 (G3) C_ Z_ R_

Aplicarea compensării razei sculei în cazul interpolării cilindrice

Comenzile G41, G42 pot fi folosite în modul obişnuit în starea activată a interpolării cilindrice. Se vor avea în vedere totuşi următoarele restricții:
- Activarea interpolării cilindrice (comanda G7.1 Qr) este posibilă numai în starea G40,
- Dacă e necesară activarea G41 sau G42 în modul de interpolare cilindrică, trebuie programat G40 înainte de a se dezactiva interpolarea cilindrică (comanda G7.1 Q0).

Restricții privind programarea în cursul interpolării cilindrice

Comenzile de mai jos nu pot fi utilizate în starea activată a interpolării cilindrice:
- selectarea planului: G17, G18, G19,
- transformările de coordonate: G52, G92,
- schimbarea sistemului de coordonate de lucru: G54, …, G59,
- orientarea în sistemul de coordonate al maşinii: G53,
- interpolarea circulară prin introducerea centrului cercului (I, J, K),
- cicluri de găurire.

Exemplu

Figura alăturată prezintă o traiectorie frezată la o adâncime de 3mm pe suprafața radială a unui cilindru cu raza R=28.65 mm. Scula rotativă T606 este paralelă cu axa X.
O deplasare de un grad pe suprafața cilindrului este:

$$28.65mm \cdot \frac{1°}{180°} \cdot \pi = 0.5mm$$

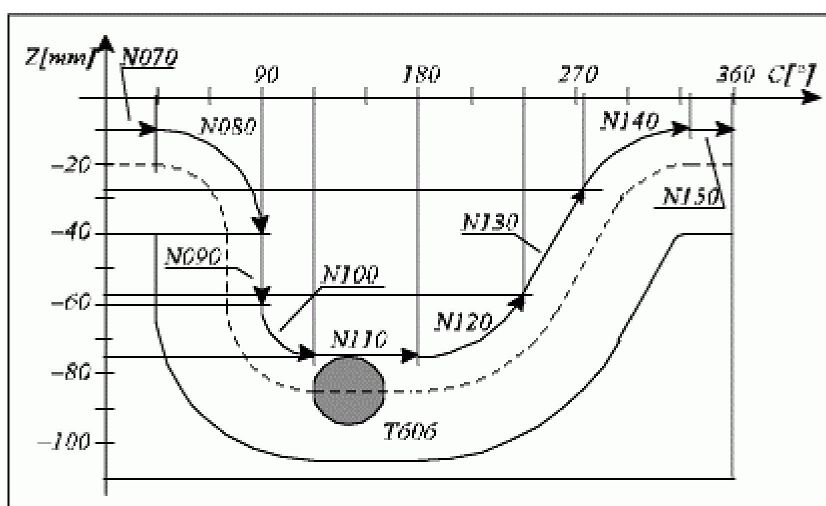


Fig. 4.7-2

Ordinea axelor care se vede în figură corespunde cu selectarea planului G19.

```
%O7602(INTERPOLARE CILINDRICĂ)
...
N020 G0 X200 Z20 S500 M3 T606
N030 G19 Z-20 C0            (G19: selectarea planului C-Z)
N040 G1 X51.3 F100
N050 G7.1 C28.65           (interpolare cilindrică activată, axa de
                            rotație :C, raza cilindrului:28.65mm)
N060 G1 G42 Z-10 F250
N070 C30
N080 G2 Z-40 C90 R30
N090 G1 Z-60
N100 G3 Z-75 C120 R15
N110 G1 C180
N120 G3 Z-57.5 C240 R35
N130 G1 Z-27.5 C275
```

```
N140 G2 Z-10 C335 R35
N150 G1 C360
N160 G40 Z-20
N170 G7.1 C0          (dezactivare interpolare cilindrică)
N180 G0 X100
...
%
```

5 Datele de intrare pentru coordonate

5.1 Programarea absolută şi incrementală (G90, G91), operatorul I

Datele de intrare ale coordonatei pot fi specificate ca valori absolute sau incrementale. Într-o specificare absolută, trebuie specificate pentru comanda numerică coordonatele punctului final, pentru datele incrementale trebuie specificată distanța de parcurs în acel bloc.

G90: programare de date absolute

G91: programare de date incrementale

G90 şi G91 sunt funcții modale. Grupul de parametri *CODES* va decide ce stare se va activa la punerea sub tensiune a sistemului.
Deplasare într-o poziție absolută se poate face doar după ce s-a efectuat întoarcere în punctul de referință.
Exemplu:
După cum se vede în figură, o deplasare se poate programa în unul din următoarele două moduri:

```
G90 G01 X20 Y50
G91 G01 X-40 Y30
```

Operatorul I va fi activ în condițiile specificării absolute a datei. Acesta se aplică doar la coordonata a cărei adresă o precede şi semnifică o dată incrementală. Modul alternativ de a rezolva exemplul de mai sus este.

```
(G90) G01 XI-40 YI30
G01 X20 YI30
G01 XI-40 Y50
```

5.2 Comanda pentru date în coordonate polare (G15, G16)

Coordonatele punctului final pot fi introduse în mod alternativ şi prin specificarea de coordonate polare, adică cu specificarea unghiului şi a razei.

G16: Comandă de date în coordonate polare

G15: Anulare comandă de date în coordonate polare

La punerea sub tensiune comanda numerică trece în starea G15. G15 şi G16 sunt funcții modale.
Datele coordonatelor polare sunt active în planul definit prin G17, G18, G19. Când este specificată o dată, adresele axelor orizontală şi verticală ale planului sunt considerate ca rază şi respectiv unghi. De exemplu, în starea G17, datele scrise la adresa X(U) şi Y(V) sunt raza şi respectiv unghiul. ATENȚIE! În starea G18, Z şi X sunt axele orizontală şi verticală (datele pentru R şi respectiv unghi).
Când este specificată o dată unghiulară, direcțiile pozitivă şi negativă pentru unghi sunt în sens orar şi respectiv antiorar.
Datele pentru celelalte axe sunt considerate date de coordonate carteziene. Raza şi unghiul pot fi specificate ca date absolute sau incrementale.
Atunci când raza este specificată ca dată absolută, originea sistemului de coordonate curent va fi originea sistemului de coordonate polare:

Exemplu:

```
G90 G16 G01 X100 Y60 F180
```

Atât raza cât şi unghiul sunt date absolute, scula se deplasează în punctul 100mm, 60° .

```
G90 G16 G01 X100 Y40 F180
```

Unghiul este dată incrementală. Se execută o deplasare de 40° față de poziția unghiulară anterioară.

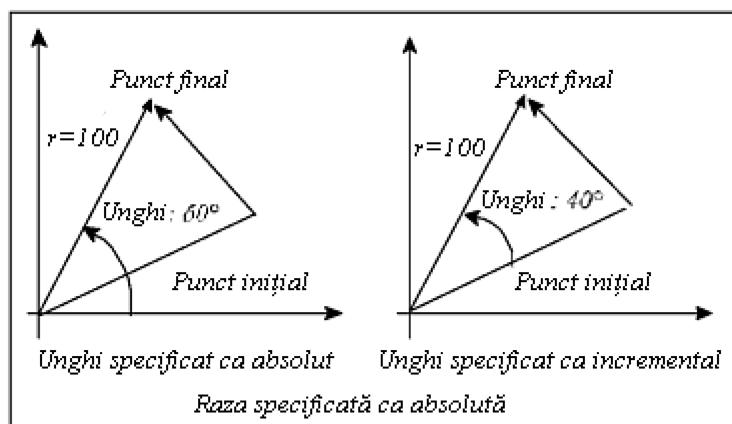


Fig. 5.2-1

Dacă raza este specificată ca valoare incrementală poziția instantanee a axelor va fi originea sistemului de coordonate polar.

Un cerc poate fi programat cu comanda pentru date în coordonate polare (G16). Cercul poate fi specificat atât prin rază cât şi prin I, J, K. În cazul al doilea comanda numerică va considera adresele I, J, K în mod invariabil drept date carteziene.

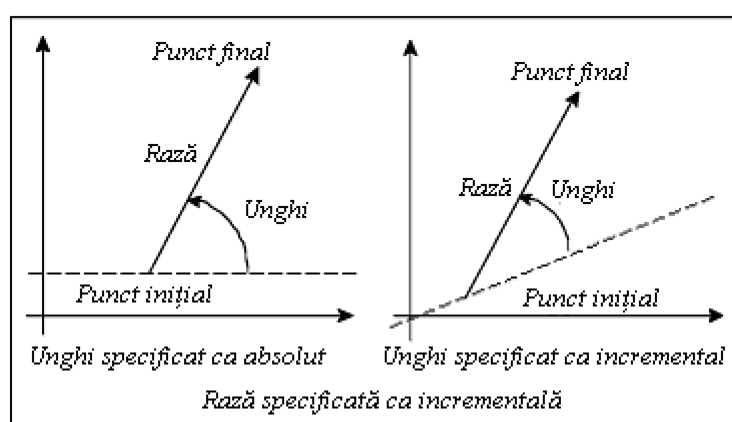


Fig. 5.2-2

Când originea sistemului de coordonate curent coincide cu centrul unui cerc sau al unei elice, se poate programa cu specificarea de coordonate polare o rotire multiplă.

Exemplu:

```
(G17 G16 G90) G02 X100 Y-990 Z50 R-100
```

În acest bloc s-a specificat o elice cu două ture în direcție antiorară. La programarea unui cerc cu mai multe ture trebuie avut în vedere că trebuie programat un unghi polar negativ sau pozitiv pentru direcțiile G2 sau respectiv G3.

Note:

Adresele folosite în următoarele instrucțiuni nu vor fi considerate ca polare:

- G 10 coordonatele folosite la impunerea instrucțiunii,
- G 52 ofset ccordonate,
- G 92 impunere coordonate,
- G 53 poziționare în sistemul de coordonate al maşinii
- G 68 rotire coordonate
- G 51 modificare la scară,
- G 50.1 imagine în oglindă.

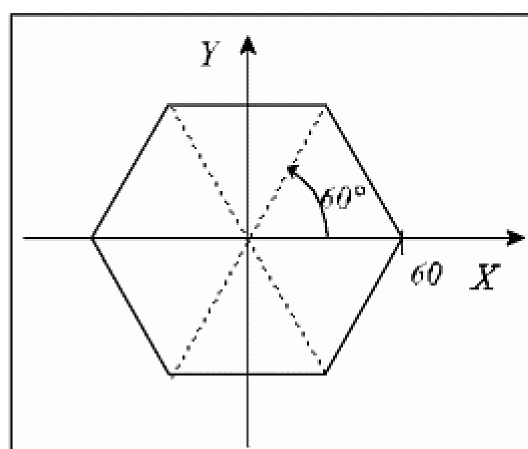


Fig. 5.2-3

Exemplu de frezare a unui hexagon:

```
N1 G90 G17 G0 X60 Y0 F120
N2 G16 G1 Y60
N3 Y120
N4 Y180
N5 Y240
N6 Y300
N7 Y360
N8 G15 G0 X100
```

5.3 Conversia inci/metric (G20,G21)

După ce s-a programat codul G potrivit, se pot specifica unităţile de măsură ale datelor de intrare:

G20: programarea introducerii în inci

G21: programarea introducerii în sistem metric

La începutul programului trebuie selectată unitatea de măsură dorită prin specificarea codului corespunzător. Unitatea de măsură selectată rămâne activată până când se introduce o comandă cu sens contrar, adică G20 şi G21 sunt coduri modale. Efectul lor se va păstra şi la deconectarea sistemului, adică unitatea de măsură care era activă înainte de decuplare va fi activă şi după reconectare.

Schimbarea unităţii de măsură va afecta următoarele elemente:

- *Coordonata şi datele de compensare,*
- *Avansul*
- *Viteza periferică constantă*
- *Afişările poziţiei, compensării şi avansului.*

5.3 Specificare şi gama de valori a datelor de coordonată

Datele de coordonată pot fi specificate cu 8 cifre zecimale.

Punctul zecimal va fi interpretat în funcţie de unitatea de măsură folosită:

- X2.134 înseamnă 2.134 mm sau 2.134 inci,
- B24.36 înseamnă 24.36 grade, atunci când axa B se referă la o axă de rotaţie.

Folosirea punctului zecimal nu este obligatorie:

- X325 înseamnă 325 mm.

Zeroul din faţa punctului zecimal se poate omite:

- .032 = 0.032

Numărul de zecimale cu care comanda numerică va interpreta un număr este funcţie de sistemul de increment folosit. De exemplu comanda X1.23456, atunci când este selectat sistemul de increment IS-B, va fi interpretată ca:

- 1.235 mm (în sistem metric)
- 1.2346 inci (în inci).

În mod corespunzător, datele de intrare vor fi transferate la ieşire ca valori rotunjite.

În tabelul de mai jos sunt prezentate valorile gamelor lungimilor coordonatelor.

Unitate de intrare	Unitate de ieşire	Sistemul de increment	Valoarea gamei lungimilor coordonatelor	Unitatea de măsură
mm	mm	IS-A	± 0.01 - 999999.99	mm
		IS-B	± 0.001 - 99999.999	
		IS-C	± 0.0001 - 9999.9999	
inci	mm	IS-A	± 0.001 – 39370.078	inci
		IS-B	± 0.0001 - 3937.0078	
		IS-C	± 0.00001 - 39370.078	
inci	inci	IS-A	± 0.001 - 99999.999	inci
		IS-B	± 0.0001 - 9999.9999	
		IS-C	± 0.00001 - 999.99999	
mm	inci	IS-A	± 0.01 - 999999.99	mm
		IS-B	± 0.001 - 99999.999	
		IS-C	± 0.0001 - 9999.9999	

Gama de valori pentru coordonatele unghiulare este cuprinsă în tabelul următor.

Sistemul de increment	Unitatea minimă care poate fi introdusă	Unitatea de măsură
IR-A	± 0.01 - 999999.99	grade
IR-B	± 0.001 - 99999.999	
IR-C	± 0.0001 - 9999.9999	

5.5 Reducerea măsurii axelor de rotaţie la 0°÷360°

Această funcţie se poate folosi în cazul axelor de rotaţie, adică dacă adresa A, B, sau C este selectată pentru operarea axei de rotaţie. Utilizarea reducerii măsurii axelor de rotaţie la 0°÷360° însemnă că poziţia unei axe nu este înregistrată între plus şi minus infinit, ci având în vedere periodicitatea fiecărei axe, adică între 0° şi 360° .

Selectarea axei de rotaţie

Selectarea se poate executa prin impunerea parametrului 0182 A.ROTARY, 0185 B.ROTARY, sau 0188 C.ROTARY pentru axa A, B, sau respectiv C. Dacă unul din aceşti parametri este impus 1:

- comanda numerică nu execută conversia inci/metric pentru axa respectivă,
- funcţia reducerii măsurii axelor de rotaţie la 0°÷360° se poate valida pentru acea axă prin impunerea parametrului ROLLOVEN corespunzător la 1.

Validarea folosirii reducerii măsurii axelor de rotaţie la 0°÷360°

Funcția este activată prin impunerea parametrului 0241 ROLLOVEN_A, 0242 ROLLOVEN_B sau 0243 ROLLOVEN_C pentru axele A, B sau respectiv C, dacă aceste axe sunt de rotație. Dacă parametrul ROLLOVEN_x este

=0 : axa de rotație este considerată ca o axă liniară şi impunerea parametrilor consecutivi nu are efect,

=1 : se aplică folosirea reducerii măsurii axelor de rotație la 0°÷360° pentru axa de rotație, a cărei semnificație se va prezenta mai jos.

Specificarea traiectoriei prin reducerea măsurii axelor de rotație la 0°÷360°

Traiectoria axei este definită pentru o rotație 0°÷360° la parametrul 0261 ROLLAMNT_A, 0262 ROLLAMNT_B sau 0263 ROLLAMNT_C în increment de intrare pentru axele A, B sau respectiv C. Astfel dacă comanda numerică lucrează în sistemul de incremente B şi axa se roteşte 360° pentru o rotație 0°÷360°, valoarea care trebuie înscrisă la parametrul respectiv este 360000.

Prin intermediul impunerii de parametri de mai sus comanda numerică va afişa întotdeauna poziția axei de rotație în gama 0° ÷ +359.999° independent de direcția de rotire şi de numărul de rotații.

Deplasarea axei de rotație în cazul programării absolute

În cazul introducerii de date absolute, când este validată folosirea reducerii măsurii axelor de rotație la 0°÷360° pentru axa de rotație (ROLLOVEN_x=1), axa nu se deplasează niciodată mai mult decât impunerea de la parametrul ROLLAMNT_x. Asta înseamnă că dacă de exemplu ROLLAMNT_C=360000 (360°), deplasare maximă este de 359.999°.
Prin impunerea parametrilor 0224 ABSHORT_A, 0225 ABSHORT_B sau 0226 ABSHORT_C, se decide dacă direcția de mişcare să fie mereu în acord cu semnul poziției introduse la adresa axei sau dacă mişcarea să se facă pe calea cea mai scurtă.
Dacă parametrul ABSHORT_x corespunzător este:
=0 : se face deplasarea întotdeauna în direcția semnului poziției programate,
=1 : se face deplasarea întotdeauna pe calea cea mai scurtă.

0188 **C.ROTARY**=1 0243 **ROLLOVEN_C**=1 0263 **ROLLAMNT_C**= =360000	Bloc programat prin introducere de coordonată absolută	Deplasarea executată prin acest bloc	Poziția la sfârşitul blocului
0246 **ABSHORT_C**=0 deplasarea se face întotdeauna în direcția semnului programat la adresa C			C=0
	G90 C450	90	C=90
	G90 C0 (**0 este un număr pozitiv!**)	270	C=0
	G90 C-90	-90	C=270
	G90 C-360	-270	C=0
0246 **ABSHORT_C**=1 deplasarea se face întotdeauna pe drumul cel mai scurt			C=0
	G90 C450	90	C=90
	G90 C0	-90	C=0
	G90 C-90	-90	C=270
	G90 C-360	90	C=0

Deplasarea axei de rotație în cazul programării incrementale

În cazul introducerii de date incrementale, direcția de mişcare este întotdeauna în acord cu semnul programat.

Parametrul corespunzător ROLLAMNT_x care să fie aplicat pentru impunerea mişcării poate fi impus la parametrul 0247 RELROUND_A, 0248 RELROUND_B respectiv 0249 RELROUND_C. Dacă parametrul corespunzător ROLLAMNT_x este

=0 : parametrul ROLLAMNT_x este dezactivat, adică deplasarea poate fi mai mare decât 360°

=1 : parametrul ROLLAMNT_x este activ. Dacă de exemplu ROLLAMNT_C=360000 (360°), deplasare maximă este de 359.999°.

0188 **C.ROTARY**=1 0243 **ROLLOVEN_C**=1 0263 **ROLLAMNT_C**= =360000	Bloc programat prin introducere de coordonată absolută	Deplasarea executată prin acest bloc	Poziția la sfârşitul blocului
0249 **ROLLAMNT _C**=0 parametrul ROLLAMNT_C este dezactivat			C=0
	G91 C450	450	C=90
	G91 C0	0	C=90
	G91 C-90	-90	C=0
	G91 C-360	-360	C=0
0249 **ROLLAMNT _C**=1 parametrul ROLLAMNT_C este activ			C=0
	G91 C450	90	C=90
	G91 C0	0	C=90
	G91 C-90	-90	C=0
	G91 C-360	0	C=0

6 Avansul

6.1 Avansul rapid

G00 comandă o poziționare cu avans rapid.
Valoarea avansului rapid pentru fiecare axă este impusă prin parametru de către constructorul maşinii. Valoarea avansului rapid poate fi diferită pentru fiecare axă.
Atunci când mai multe axe se deplasează simultan cu avans rapid, rezultanta avansului va fi calculată astfel încât componenta vitezei pe fiecare axă să nu depăşească valoarea particulară a avansului rapid pentru acea axă (impusă ca parametru) şi poziționarea să se realizeze în minimul de timp.
Valoarea avansului rapid este modificată cu comutatorul de corecție a avansului rapid care poate avea treptele:

F0: definit prin parametrul RAPOVER în %,
şi 25%, 50%, 100%.

Valoarea avansului rapid nu poate depăşi 100%.
Deplasarea cu avans rapid se va opri dacă comutatorul de corecție avans este pe poziția 0%.
În lipsa unui punct de referință valid, sunt valabile valorile reduse ale avansurilor rapide definite de constructorul maşinii pentru fiecare axă până când se execută întoarcerea în punctul de referință.
Valorile de corecție ale avansului rapid pot fi conectate cu cele ale comutatorului de corecție avans.
Dacă se deplasează axele cu butoanele de JOG, viteza de avans rapid este diferită față de avansul rapid cu G00, şi de asemeni se poate selecta prin parametri pentru fiecare axă.
De obicei viteza de deplasare cu butoanele de JOG este mai mică decât viteza de poziționare şi corelată cu timpul de răspuns al omului.

6.2 Valoarea avansului de prelucrare

Avansul este programat la adresa **F**.
Avansul programat este realizat în blocuri de interpolare liniară (G01) şi circulară (G02, G03).
Avansul este realizat tangențial de-a lungul traiectoriei programate.

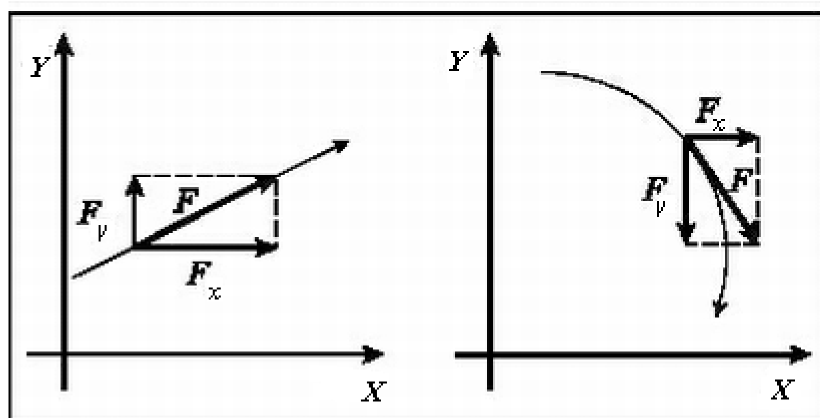


Fig. 6.2-1

F – avans tangențial (valoare programată)
F_x – componentă a avansului pe direcția X
F_y - componentă a avansului pe direcția Y

$$F=\sqrt{F_x^2+F_y^2}$$

Cu excepția stărilor de inhibare corecție şi stop (G63), valoarea avansului programat poate fi modificată în gama de la 0 la 120% cu comutatorul de corecție a avansului.

Valoarea avansului (F) este modală. După punerea sub tensiune va fi activă valoarea avansului impusă cu parametrul *FEED*.

6.2.1 Avansul pe minut (G94) şi avansul pe rotaţie (G95)

Unitatea de avans poate fi specificată în program cu codurile G94 şi G95:

G94: avansul pe minut
G95: avansul pe rotaţie

Termenul „avans/minut“ se referă la avansul specificat în unităţile mm/min, inci/min sau grade/min.
Termenul „avans/rotaţie“ se referă la avansul realizat într-o rotaţie a arborelui, specificat în unităţile mm/rot, inci/rot sau grade/rot. Nu poate fi programat G95 decât dacă arborele este echipat cu un codor de poziţie.
G94 şi G95 sunt valori modale. După punerea sub tensiune va fi selectată una dintre stările G94 sau G95 funcţie de parametrii din grupul *CODES.* Avansul rapid se execută invariabil în mm/min, deci starea G94/G95 nu îl va afecta.

Tabelul următor prezintă gama maximă a valorilor programabile la adresa F, pentru diferite cazuri.

Unităţi de intrare	Unităţi de ieşire	Sistem de increment	Gama valorii la adresa F	Unitate
mm	mm	IS-A	0.001 - 250000	mm sau grade/min
		IS-B	0.0001 – 25000	
		IS-C	0.00001 – 2500	
		IS-A	0.0001 – 5000	mm sau grade/rot
		IS-B	0.00001 – 500	
		IS-C	0.000001 – 50	
inci	mm	IS-A	0.0001 – 9842.5197	inci sau grade/min
		IS-B	0.00001 - 984.25197	
		IS-C	0.000001 - 98.425197	
		IS-A	0.00001 – 196.85039	inci sau grade/rot
		IS-B	0.000001 – 19.685039	
		IS-C	0.0000001 – 1.9685039	
inci	inci	IS-A	0.0001 – 25000	inci sau grade/min
		IS-B	0.00001 – 2500	
		IS-C	0.000001 – 250	
		IS-A	0.00001 – 500	inci sau grade/rot
		IS-B	0.000001 – 50	
		IS-C	0.0000001 – 5	
mm	inci	IS-A	0.001 - 250000	mm sau grade/min
		IS-B	0.0001 – 25000	
		IS-C	0.00001 – 2500	
		IS-A	0.0001 – 5000	mm sau grade/rot
		IS-B	0.00001 – 500	
		IS-C	0.000001 – 50	

6.2.2 Limitarea superioară a avansului de prelucrare

Valoarea maximă pentru avansul de prelucrare la o anumită maşină, poate fi limitată superior (impusă ca un parametru) de către constructor. Valoarea impusă aici se referă invariabil la avansul în mm/min. Această valoare este de asemenea şi viteza de DRY RUN (executarea programului fără prelucrare). În cursul executării unui program, dacă valoarea avansului programat depăşeşte valoarea limită superioară, comanda numerică o va limita la aceasta.

De asemenea, separat, se poate limita superior valoarea maxim a avansului pentru JOG prin parametrii pentru timpii de răspuns uman.

6.3 Accelerarea/frânarea automată

La deplasarea cu avans rapid, comanda numerică va executa în mod automat o accelerare liniară la începutul mişcării şi o frânare liniară la sfârşitul mişcării. Mărimea acceleraţiei este definită de către constructorul maşinii în parametrul *ACCn,* în funcţie de dinamica maşinii.

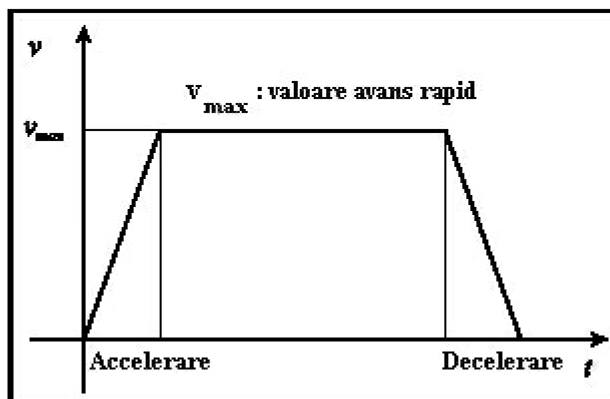


Fig. 6.3-1

În mişcările de avans, valoarea avansului tangenţial (programat) va fi luată în calcul de către comanda numerică pentru acceleraţia liniară, şi respectiv, valoarea sa va fi scăzută prin frânarea liniară. Această tehnică oferă avantajul faţă de accelerarea tradiţională (exponenţială), că maşina va atinge mai rapid viteza dorită (considerând o anumită constantă de timp adoptată în ambele cazuri). Astfel timpii de accelerare şi de frânare (adică timpii de deplasare efectivă a axelor) vor fi reduşi.

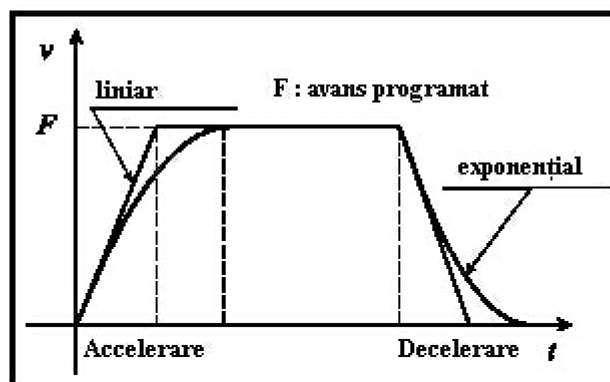


Fig. 6.3-2

Un alt avantaj al accelerării liniare faţă de accelerarea exponenţială este o mai mică distorsionare a profilului (adică eroarea de rază), comparativ cu cea obţinută la accelerarea exponenţială, la prelucrarea cu viteză mare a unui cerc.

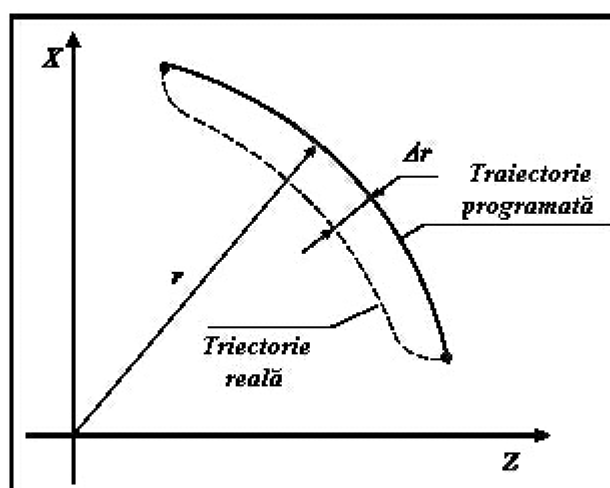


Fig. 6.3-3

Comanda numerică monitorizează modificările vitezelor tangenţiale. Aceasta este necesar pentru a atinge viteza comandată într-un proces de accelerare continuă, dacă este necesar, prin câteva blocuri. Accelerarea la noul avans (mai mare decât cel precedent) este începută de către comanda numerică în mod invariabil în execuţia blocului respectiv, în care este specificată noua valoare a avansului. Dacă este necesar, acest proces se poate prelungi pe câteva blocuri. Frânarea la o valoare nouă a avansului (mai mică decât cea precedentă) va fi începută de către comanda numerică în blocul precedent celui în cauză, astfel ca prelucrarea să pornească cu viteza programată în blocul în care este specificată noua valoare a avansului. Accelerarea /frânarea liniară este folosită şi la deplasarea manuală în modul JOG sau cu roata de mână. Valorile pentru acest caz vor fi definite pentru fiecare axă prin parametrii de la *ACC1* la *ACC8*.

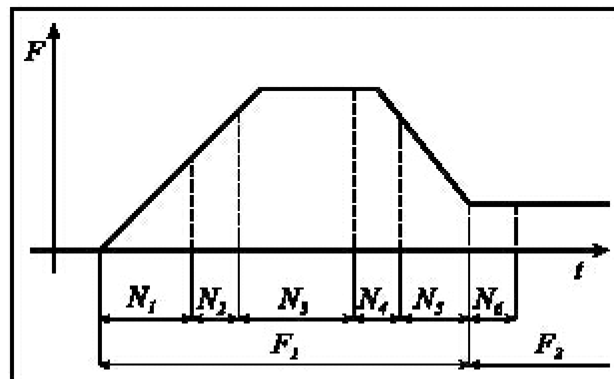


Fig. 6.3-4

6.4 Funcţiile de control a avansului

Funcţiile de control a corecţiei sunt necesare atunci când trebuie prelucrate colţuri, şi/sau când o tehnologie particulară necesită anularea comutatorului de corecţie şi a butonului de stop. Când se prelucrează colţuri cu aplicarea tăierii continue, axele, datorită inerţiei lor, nu sunt capabile să urmărească traiectoria comandată de către comanda numerică. Astfel, scula va rotunji colţurile, mai mult sau mai puţin, funcţie de avans. Dacă piesa de prelucrat necesită colţuri ascuţite, trebuie specificată reducerea avansului la sfârşitul blocului, să se aştepte până când axele se opresc, şi numai după aceea să se pornească următoarea mişcare.

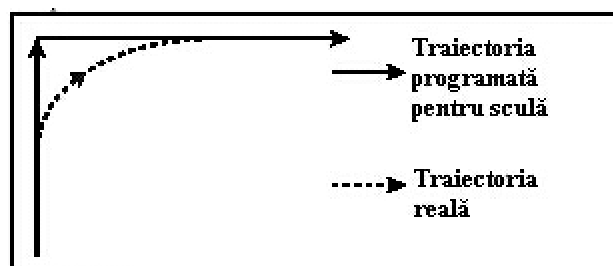


Fig. 6.4-1

6.4.1 Oprirea precisă (G09)

G09 nu este o funcţie modală, deci nu este activă decât în blocul în care a fost programat. La sfârşitul blocului în care a fost specificat G09, comanda numerică încetineşte după executarea interpolării, şi aşteaptă apariţia semnalului „atingere poziţie". Dacă acest semnal nu este primit în 5 secunde, se afişează mesajul *1020 POSITION ERROR (eroare de poziţie).* Această funcţie poate fi utilizată pentru prelucrarea precisă a colţurilor ascuţite.

6.4.2 Modul tăiere precisă (G61)

Este o funcţie modală anulată cu comenzile G62, G63 sau G64.
Comanda numerică va încetini până la finalizarea fiecărei interpolări şi va aştepta apariţia semnalului „atingere poziţie". Următoarea interpolare va fi pornită numai după apariţia

semnalului Dacă acest semnal nu este primit în 5 secunde, se afişează mesajul *1020 POSITION ERROR (eroare de poziţie).*

6.4.3 Modul tăiere continuă (G64)

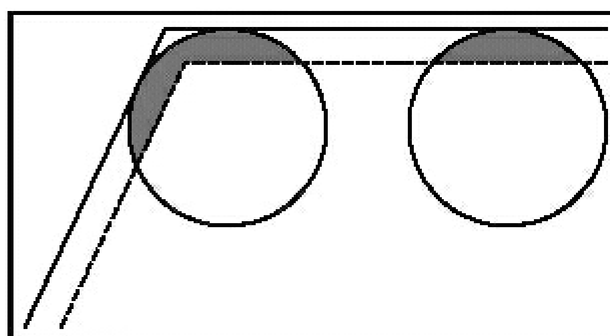
Funcţie modală. Comanda numerică va activa această stare la punerea sub tensiune. Ea va fi anulată de unul din codurile G61, G62 sau G63.
În acest mod nu se aşteaptă terminarea interpolării, axele nu vor încetini. În loc de aceasta, va fi începută imediat interpolarea următorului bloc.
În acest mod nu pot fi prelucrate colţuri ascuţite, deoarece ele vor fi rotunjite.

6.4.4 Modul inhibare corecţie şi stop (filetare) (G63)

Este o funcţie modală anulată cu comenzile G61, G62 sau G64.
În acest mod sunt inhibate corecţiile pentru avans şi arbore şi stopul avansului. Valorile corecţiilor sunt considerate 100% (indiferent de poziţia comutatorului). La terminarea interpolării, comanda numerică nu va încetini, ci va începe imediat următorul ciclu de interpolare.
Acest mod este aplicabil la filetarea cu cuţit sau cu tarod.

6.4.5 Corecţia automată la colţ (G62)

Este o funcţie modală anulată cu comenzile G61, G62 sau G64.
Atunci când se prelucrează interiorul unui colţ, asupra sculei acţionează forţe mai mari înainte şi după colţ. Pentru a preveni supraîncărcarea sculei şi apariţia vibraţiilor, comanda numerică - atunci când este comandat G62 - va reduce în mod automat avansul înainte şi după un colţ interior.


Fig. 6.4.5-1

Corecţia de colţ este activă în următoarele condiţii:
- când este activă compensarea sculei (G41, G42).
- între blocurile G0, G1, G2, G3.
- în deplasările în planul selectat.
- când este prelucrat interiorul colţului.
- când unghiul unui colţ este mai mic decât un anumit unghi definit printr-un parametru.
- pe o distanţă înainte şi după colţ, definită prin parametri.

Funcţia de corecţie de colţ va fi activă între fiecare din următoarele perechi de blocuri: liniar la liniar, liniar la circular, circular la liniar şi circular la circular.
Interiorul unghiului Θ poate fi selectat între 1 şi 180° prin parametrul *CORNANGLE.*

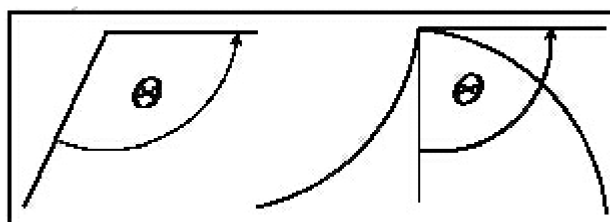


Fig. 6.4.5-2

Frânarea şi accelerarea vor fi începute la distanţele L_1 şi L_g înainte şi respectiv după colţ. În cazul arcelor de cerc, distanţele L_1 şi L_g, vor fi calculate de către comanda numerică de-a lungul arcului. Distanţele L_1 şi L_g vor fi definite în parametrii *DECDIST* şi respectiv *ACCDIST*

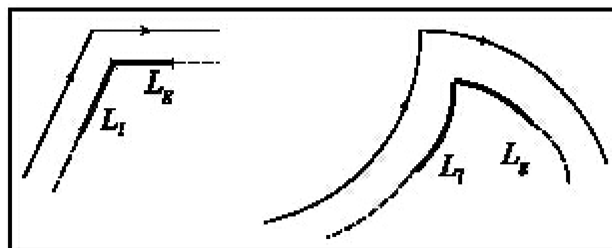


Fig. 6.4.5-3

Valoarea corecţiei poate fi selectată ca un procent în parametrul *CORNOVER*.. Corecţia va deveni activă la distanţa L_1 înainte de colţ şi va continua să fie activă pe distanţa L_g după colţ. Comanda numerică va ţine cont atât de corecţia avansului cât şi de corecţia de colţ.:

F* corecţia avansului * corecţia de colţ.

G09 se scrie într-un anumit bloc pentru a programa o oprire precisă în starea G62.

6.4.6 Corecţia tăierii circulare interioare

Dacă este activă compensarea sculei (G41, G42) comanda numerică va reduce automat avansul de prelucrare la interiorul suprafeţei unui arc astfel ca avansul să aibă valoarea programată de-a lungul razei de tăiere. Avansul în centrul razei sculei este

$$F_c = \frac{R_c}{R} F$$

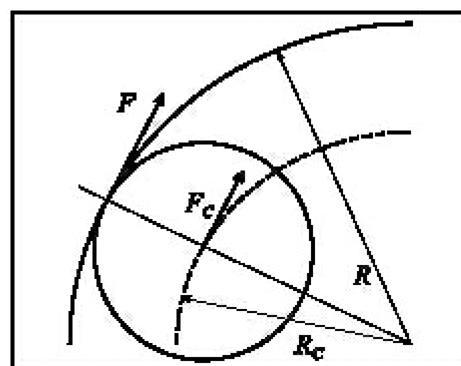


Fig. 6.4.6-1

Unde:

F_c este avansul corectat al centrului razei sculei

R este raza cercului programată

R_c este raza cercului corectată

F este avansul programat.

Limita inferioară a reducerii automate a avansului este impusă prin parametrul *CIRCOVER*, în care poate fi specificată ca procent corecţia minimă. Înainte de a se transmite mai departe, corecţia pentru raza cercului este multiplicată cu valorile corecţiei avansului şi corecţiei de colţ.

7 Întârzierea (G04)

Comanda

(G94) **G04** P…

va programa o întârziere în secunde.
Domeniul pentru P este de la 0.001 la 99999.999 secunde.

Comanda

(G95) **G04** P…

va programa o întârziere în număr de rotații ale arborelui.
Domeniul pentru P este de la 0.001 la 99999.999 rotații.
Funcție de parametrul *SECOND*, întârzierea poate fi considerată în secunde indiferent de starea G94 sau G95.
Întârzierea implică invariabil întârzierea executării următorului bloc. Ea nu este o funcție modală.
Pe durata întârzierii, se afişează pe ecran, în câmpul 5 care indică interpolarea, mesajul DWL care atrage atenția operatorului de ce s-a oprit prelucrarea.

8 Punctul de referinţă

Punctul de referinţă este o poziţie distinctă la o maşină unealtă, în care se poate reveni cu uşurinţă. Poziţia punctului de referinţă poate fi definită ca un parametru în sistemul de coordonate al maşinii.
Sistemul de coordonate al piesei poate fi măsurat şi se poate face poziţionarea absolută după executarea întoarcerii în punctul de referinţă. Poziţiile limitelor de cursă introduse prin parametri şi funcţia de verificare a cursei sunt active numai după întoarcerea în punctul de referinţă.

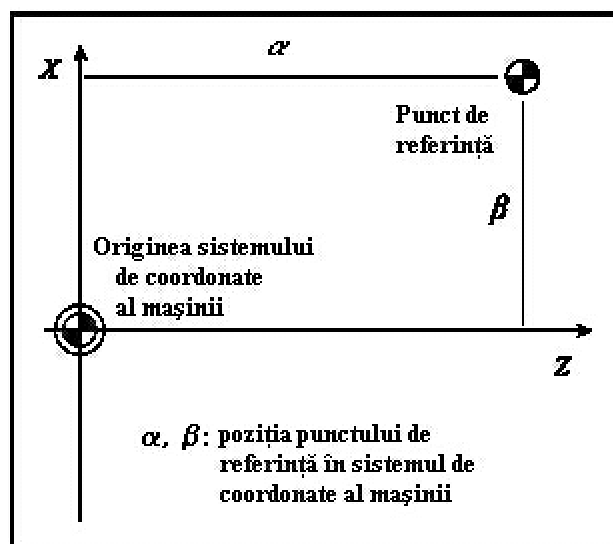


Fig. 8-1

8.1 Revenirea automată în punctul de referinţă (G28)

Instrucţiunea

G28 v

va aduce axele definite de vectorul v în punctul de referinţă. Mişcările se fac în două etape. La început se vor deplasa axele cu interpolare liniară şi avans rapid în coordonatele intermediare definite prin vectorul v. Coordonatele specificate pot fi valori absolute sau incrementale. Deplasarea se face invariabil în sistemul de coordonate curent.
Când se atinge punctul final al deplasării liniare, este şters vectorul de compensare a sculei. Coordonatele punctului intermediar vor fi stocate pentru axele definite de vectorul v.
În etapa a doua se va executa deplasarea din punctul intermediar în punctul de referinţă, simultan pe fiecare axă definită de vectorul v. Întoarcerea în punctul de referinţă este executată cu mişcare neliniară la o viteză definită pentru fiecare axă. În continuare, similar cu întoarcerea manuală, poziţia va fi acceptată în maniera definită de parametri.
Acesta nu e un cod modal.

Note:

- În afara cazului în care există un punct de referinţă valid, în comanda G28 trebuie date valori incrementale pentru coordonatele intermediare v.
- Dacă într-un bloc este programat G28, coordonatele intermediare v vor fi stocate până la deconectarea tensiunii. Cu alte cuvinte, valoarea intermediară definită într-o comandă G28 anterioară va fi în continuare activă pentru coordonatele cărora nu le-au fost date valori în comanda G28 curentă. Exemplu:

  G28 X100 punct intermediar: X=100, Y=0
  G28 Y200 punct intermediar: X=100, Y=200

8.2 Revenirea automată în punctele de referință 2, 3, 4 (G30)

Seria de instrucțiuni

G30 v P

va trimite axele de coordonate definite la adresele vectorului v în punctul de referință definit la adresa P.

P1 = punctul de referință 1
P2 = punctul de referință 2
P3 = punctul de referință 3
P4 = punctul de referință 4

Punctele de referință sunt poziții speciale definite prin parametri (REFPOS1, …, REFPOS4) în sistemul de coordonate al maşinii, folosite pentru pozițiile de schimbare, de exemplu poziții pentru schimbare de sculă. Primul punct de referință este în mod invariabil poziția punctului de referință al maşinii, adică punctul în care se face deplasarea atunci când se face întoarcerea în punctul de referință.
Instrucțiunea se poate aplica doar după ce s-a executat întoarcerea în punctul de referință.
Mişcarea se face în două etape. La început se vor deplasa axele cu mişcare liniară şi avans rapid în coordonatele intermediare definite prin vectorul v. Coordonatele specificate pot fi valori absolute sau incrementale. Deplasarea se face invariabil în sistemul de coordonate curent. Când se atinge punctul final al deplasării liniare, este şters vectorul de compensare a sculei. Coordonatele punctului intermediar vor fi stocate în sistemul de coordonate curent pentru axele definite de vectorul v. Stocate în acest fel, coordonatele se vor suprascrie peste cele stocate în instrucțiunea G28.
În a doua etapă, axele definite prin vectorul v se vor deplasa cu avans rapid din punctul intermediar în punctul de referință selectat la adresa P.
Întoarcerea în punctul de referință se face fără a se ține seama de vectorii de compensare (lungime, ofset, ofseturi tridimensionale) care nu e nevoie să fie şterşi înainte de a se introduce instrucțiunea G30, dar vor fi implementați de către comanda numerică atunci când sunt programate următoarele mişcări. În primul bloc de mişcare se restabileşte automat compensarea sculei.
Nu este un cod modal.

8.3 Revenirea automată din punctul de referință (G29)

Instrucțiunea

G29 v

va comanda întoarcerea din punctul de referință pe axele definite în vectorul v. Urmând după G28 şi G30, comanda G29 va fi executată în aceeaşi manieră. Revenirea este îndeplinită în două etape.
În prima etapă se va face deplasarea din punctul de referință în punctul intermediar înregistrat în timpul execuției instrucțiunii G28 sau G30, pentru axele definite prin vectorul v. Coordonatele punctului intermediar sunt modale, cu alte cuvinte, comanda numerică va lua în considerație valorile anterioare dacă se face referința pe o axă la care nu s-au transferat coordonate în blocul G28 sau G30 care precede G29. Se va face deplasarea în punctul intermediar ținând cont de lungimea sculei, ofsetul sculei şi compensarea tridimensională a razei sculei. Coordonatele punctului intermediar sunt eficace invariabil în sistemul de coordonate al piesei curente. În mod corespunzător, dacă de exemplu s-a programat o schimbare a sistemului de coordonate al piesei după întoarcerea în punctul de

referință şi înainte de instrucțiunea G29, punctul intermediar va fi luat în considerație în noul sistem de coordonate.

În a doua etapă se va deplasa din punctul intermediar în punctul v definit în instrucțiunea G29. Dacă coordonata v are o valoare incrementală, deplasamentul va fi măsurat din punctul intermediar.

Atunci când este impusă compensarea sculei, deplasarea în punctul final se va face ținând cont de vectorul de compensare.

Codul nu este modal.

Exemplu de utilizare a G30 şi G29:

```
...
G90
...
G30 P1 X500 Y200
G29 X700 Y150
...
...
```

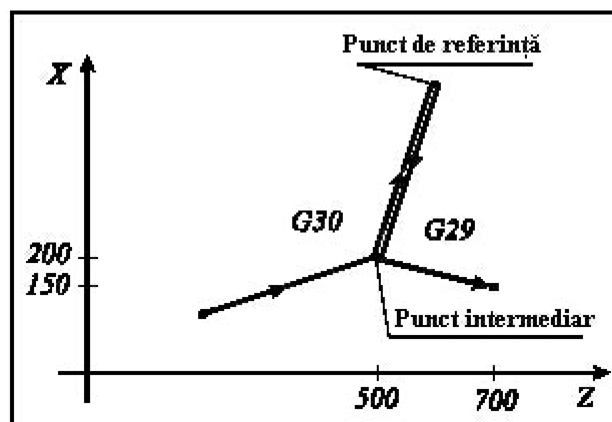


Fig. 8.3-1

9 Sisteme de coordonate, Selectarea planului

Poziția în care trebuie adusă scula este specificată în program prin data coordonatei. Când sunt disponibile trei axe (X, Z, Z), poziția sculei este exprimată prin trei date de coordonată X___ Y___ Z___ :

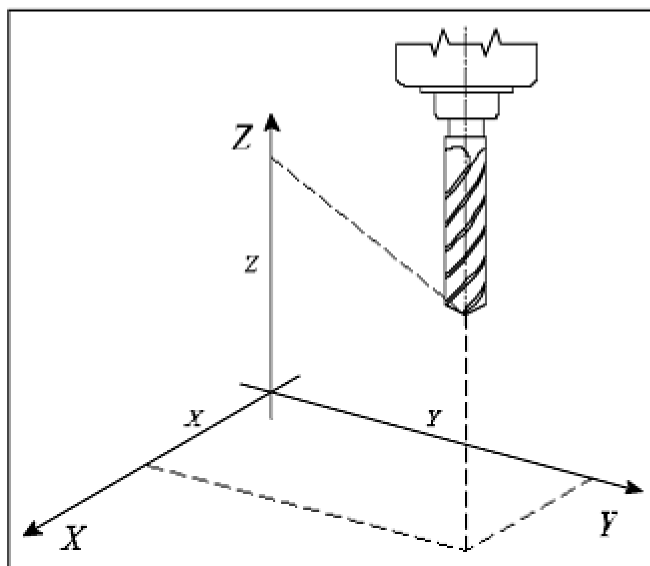


Fig. 9-1

Poziția sculei este exprimată cu atâtea date de coordonată diferite, câte axe are maşina. Datele de coordonată se referă invariabil la un sistem de coordonate dat.
Comanda numerică va face diferență între trei sisteme de coordonate diferite:

1. sistemul de coordonate al maşinii
2. sistemul de coordonate al piesei
3. sistemul de coordonate local

9.1 Sistemul de coordonate al maşinii

Punctul de zero al maşinii, adică originea sistemului de coordonate al maşinii, este un punct de pe maşina unealtă, care de obicei este definit de către constructor. Comanda numerică va defini sistemul de coordonate al maşinii în momentul întoarcerii în punctul de referință. După ce a fost definit sistemul de coordonate al maşinii, el nu va mai fi modificat de schimbarea sistemului de coordonate de lucru (G54… G59) sau de altă transformare de coordonate (G52, G92), ci doar de oprirea comenzii numerice.

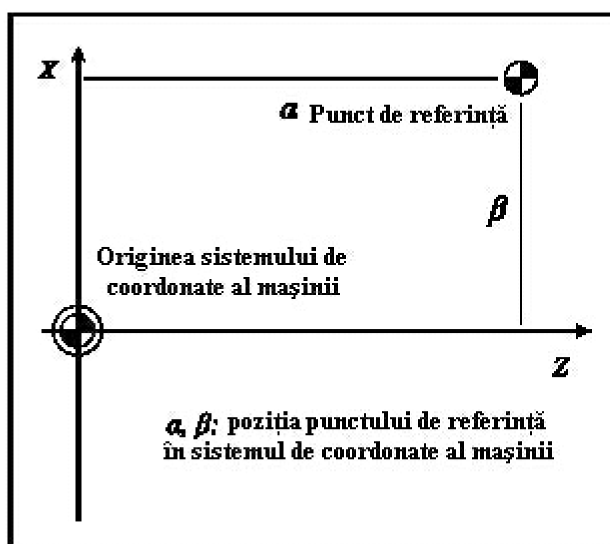


Fig. 9.1-1

9.1.1 Impunerea sistemului de coordonate al maşinii

După o revenire în punctul de referinţă, sistemul de coordonate al maşinii poate fi impus în parametri. Distanţa la punctul de referinţă, calculată de la originea sistemului de coordonate al maşinii, trebuie scrisă ca parametru.

9.1.2 Poziţionarea în sistemul de coordonate al maşinii (G53)

Instrucţiunea

G53 v

va deplasa scula în poziţia coordonatei v în sistemul de coordonate al maşinii
-Indiferent de stările G90, G91, coordonatele v sunt considerate întotdeauna drept coordonate absolute,
- operatorul I nu este activ atunci când este plasat înaintea adresei coordonatei,
- similar cu instrucţiunea G90, deplasările se fac cu avans rapid,
- poziţionarea se execută invariabil ţinându-se cont de compensarea lungimii sculei selectate.

O instrucţiune G53 poate fi executată numai după efectuarea întoarcerii în punctul de referinţă. G53 este o instrucţiune care e valabilă doar în blocul în care se află.

9.2 Sistemul de coordonate de lucru

Sistemul de coordonate aplicat la prelucrarea piesei este numit „sistemul de coordonate de lucru ". În comanda numerică pot fi definite şase sisteme diferite de coordonate de lucru.

9.2.1 Impunerea sistemului de coordonate de lucru

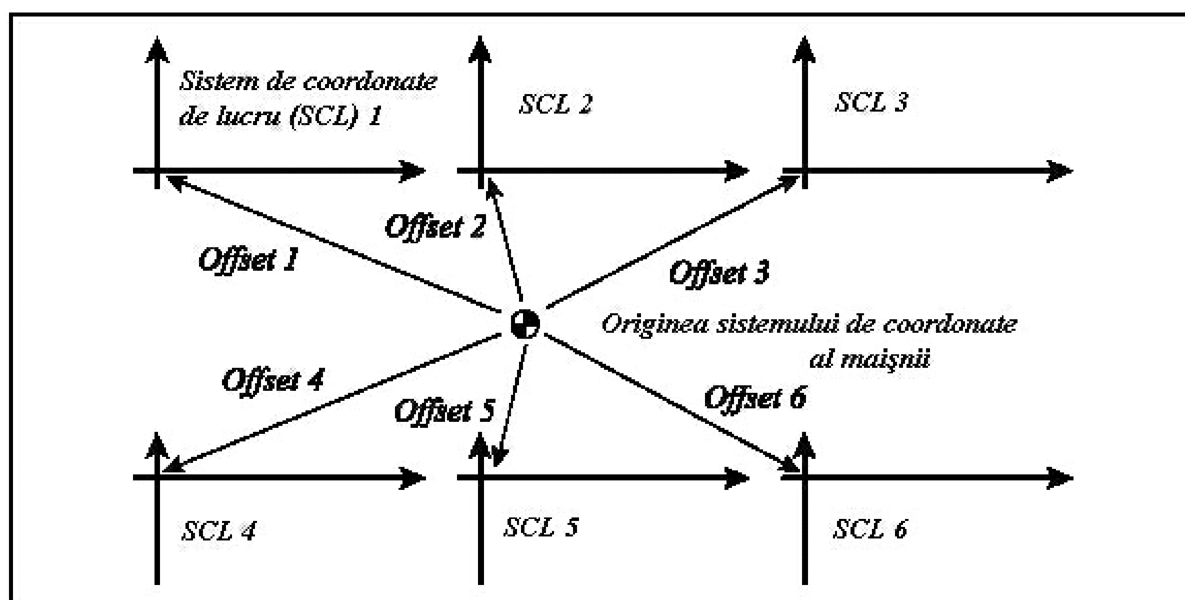


Fig. 9.2.1-1

În modul de impunere se pot stabili locaţiile diferitelor sisteme de coordonate de lucru în sistemul de coordonate al maşinii, şi se pot face ofseturile corespunzătoare.

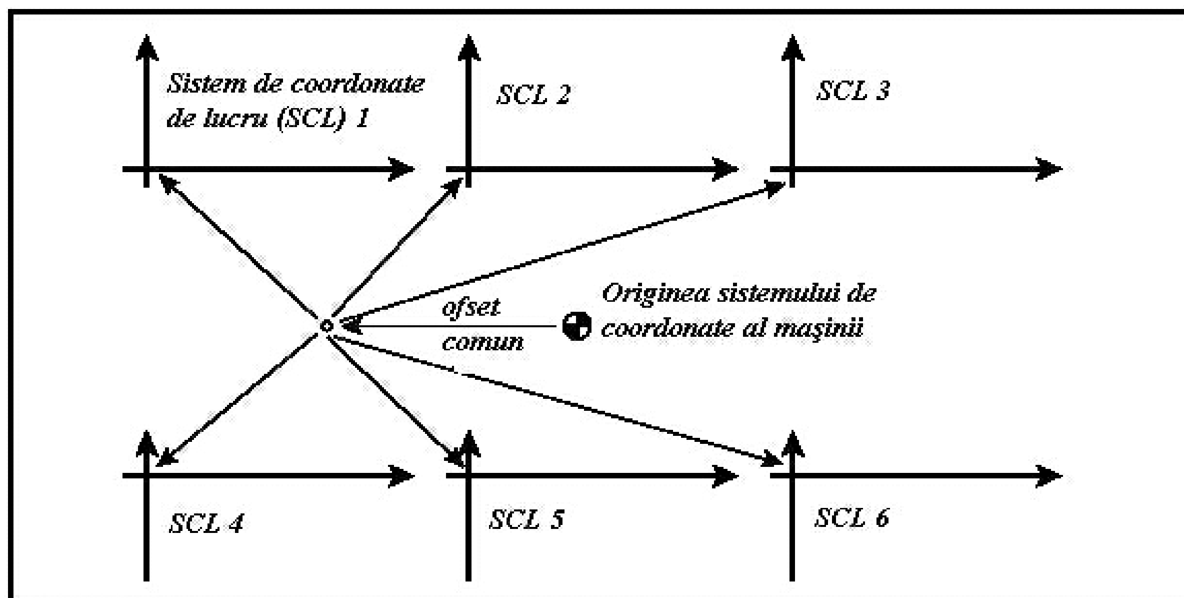


Fig. 9.2.1-2

Mai mult, se poate impune un ofset cu valoare comună pentru toate sistemele de coordonate de lucru. Aceasta se face tot în modul de impunere.

9.2.2 Selectarea sistemului de coordonate de lucru

Se poate selecta unul dintre sistemele de coordonate de lucru cu instrucțiunile G54… G59:

G54……. sistem de coordonate de lucru 1
G55……. sistem de coordonate de lucru 2
G56……. sistem de coordonate de lucru 3
G57……. sistem de coordonate de lucru 4
G58……. sistem de coordonate de lucru 5
G59……. sistem de coordonate de lucru 6

Acestea sunt funcții modale. Selectarea lor înainte de o întoarcere în punctul de referință nu are efect. După o întoarcere în punctul de referință, va fi selectat automat sistemul de coordonate de lucru 1 (G54).

Datele de coordonată absolută ale blocurilor de interpolare vor fi considerate de către comanda numerică în sistemul de coordonate de lucru curent.

De exemplu instrucțiunea

```
G56 G90 G00 X60 Y40
```

va deplasa sistemul în punctul X=60, Y=40 al sistemului de coordonate de lucru 3.

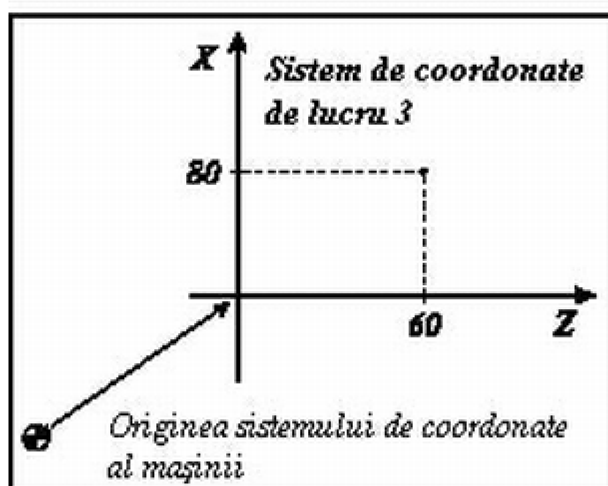


Fig. 9.2.2-1

După o schimbare a sistemului de coordonate de lucru, poziția sculei va fi afişată în noul sistem de coordonate. Să considerăm că pe masa maşinii sunt două piese. Primul sistem de coordonate de lucru (G54) este ataşat punctului de zero al uneia din piese, care are un ofset X=300, Y=800 (calculat în sistemul de coordonate al maşinii). Al doilea sistem de coordonate de lucru (G55) este ataşat punctului de zero al piesei a doua, care are un ofset X=1300, Y=400 (calculat în sistemul de coordonate al maşinii). Poziția sculei este X'=700, Y'=500 în sistemul de coordonate X', Y' (G54). Ca rezultat al instrucțiunii G55, poziția sculei va fi interpretată în sistemul de coordonate X", Y" (X"=-300, Y"= 900).

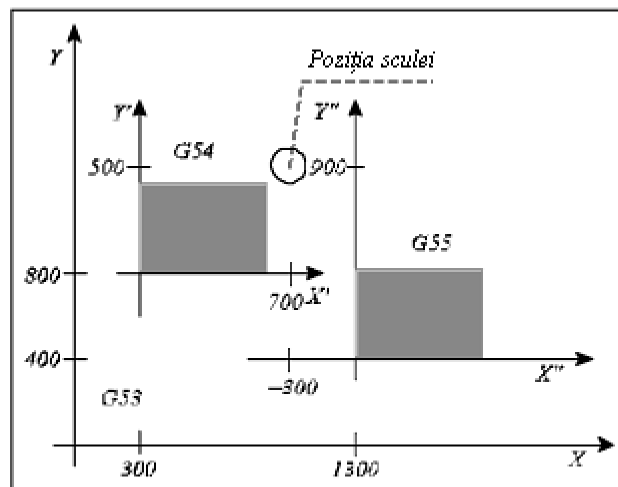


Fig. 9.2.2-2

9.2.3 Impunerea programată a ofsetului punctului de zero de lucru

De asemeni, se poate impune prin programare sistemul de coordonate de lucru şi ofsetul comun pentru acesta.
Aceasta se realizează cu instrucțiunea

G10 v **L2** Pp

unde

p = 0 impune ofsetul comun
p = 1…6 selectează sistemul de coordonate de lucru 1…6
v = ofset pentru fiecare axă

Datele coordonatelor sunt introduse în mod invariabil ca valori absolute rectangulare (carteziene). Funcția G10 nu este modală.

9.2.4 Crearea unui nou sistem de coordonate de lucru (G92)

Instrucțiunea

G92 v

va stabili un sistem nou de coordonate de lucru astfel încât punctul de coordonată v al noului sistem va fi un punct selectat – de exemplu vârful sculei (dacă este programată o compensare de lungime) sau punctul de bază al suportului sculei (dacă lipseşte compensarea de lungime). În continuare orice comandă absolută suplimentară se va referi la noul sistem de coordonate, iar pozițiile vor fi afişate în acest sistem de coordonate. Coordonatele specificate în comanda G92 vor fi interpretate întotdeauna ca valori rectangulare absolute.

Dacă, de exemplu scula este la punctul de coordonată X=150, Y=100, în sistemul de coordonate X, Y actual (curent), instrucţiunea

```
G92 X90 Y60
```

va crea un sistem de coordonate nou X', Y' în care scula se va afla în punctul de coordonată X'=90, Y'=60.
Componentele axiale ale vectorului de ofset v' dintre coordonatele sistemelor X, Y şi X', Y' sunt

$v'_x=150-90=60$

şi

$v'_y=100-60=40$.

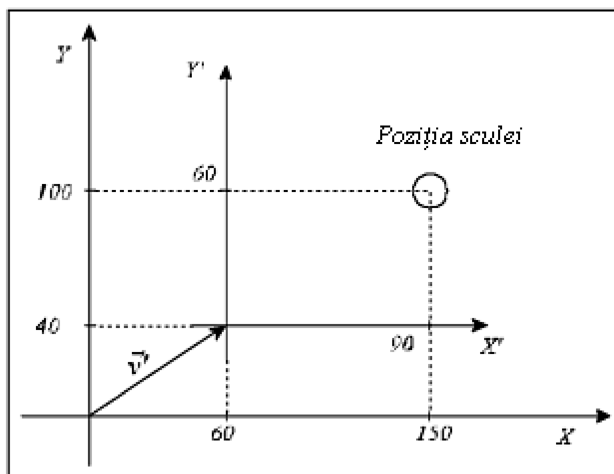


Fig. 9.2.4-1

Comanda G92 va prevala în fiecare din cele şase sisteme de coordonate de lucru, adică un ofset v calculat pentru unul din ele va fi luat în seamă şi pentru celelalte.

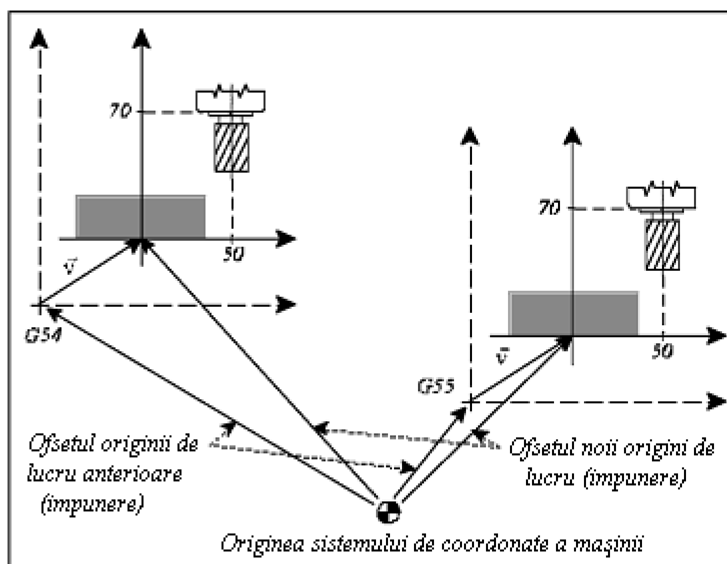


Fig. 9.2.4-2

Note:

- ofsetul sistemului de coordonate de lucru impus cu instrucţiunea G92 va fi şters prin execuţia instrucţiunilor de „sfârşit de program" (M2, M30) şi prin resetarea programului.
- instrucţiunea G92 va şterge datele ofseturilor sistemului de coordonate local (programat cu instrucţiunea G52) pe axele incluse în instrucţiune.
- Instrucţiunea G92 oferă o cale comodă pentru indicarea poziţiei ciclice a indexării mesei rotative care execută câteva rotaţii. Dacă de exemplu axa B a fost rotită în poziţia 360°, axa poate fi deplasată în poziţia 0° fără să se efectueze o mişcare fizică prin programarea G92 B0.

9.3 Sistemul de coordonate local

Atunci când se scriu programele-piesă, este mult mai la îndemână să se specifice datele coordonatelor într-un sistem de coordonate „local" în locul sistemului de coordonate de lucru.

Instrucțiunea:

G52 v

va crea un sistem de coordonate local.

- Dacă coordonata v este specificată ca o valoare absolută, originea sistemului de coordonate local va coincide cu punctul v în sistemului de coordonate de lucru.
- Când se specifică drept valoare incrementală, originea sistemului de coordonate local se va deplasa cu ofsetul v (considerând sistemul de coordonate local ca definit anterior, ori altfel ofsetul este considerat față de originea sistemului de coordonate de lucru).

În continuare, orice comandă de mişcare specificată în coordonate absolute va fi executată în noul sistem de coordonate. De asemeni, pozițiile sunt afişate în noul sistem de coordonate. Valorile coordonatelor v vor fi considerate invariabil drept coordonate carteziene.

Dacă, de exemplu, scula se află în punctul de coordonate X=150,Y=100 în sistemul de coordonate de lucru X, Y curent, instrucțiunea

```
G90 G52 X60 Y40
```

va crea un nou sistem de coordonate local X', Y' în care coordonatele sculei vor fi X'=90, Y'=60. Instrucțiunea G52 este folosită pentru definirea componentelor axiale ale vectorului de ofset v' dintre sistemele de coordonate X, Y şi X', Y' (v'_x=80, v'_y=60).

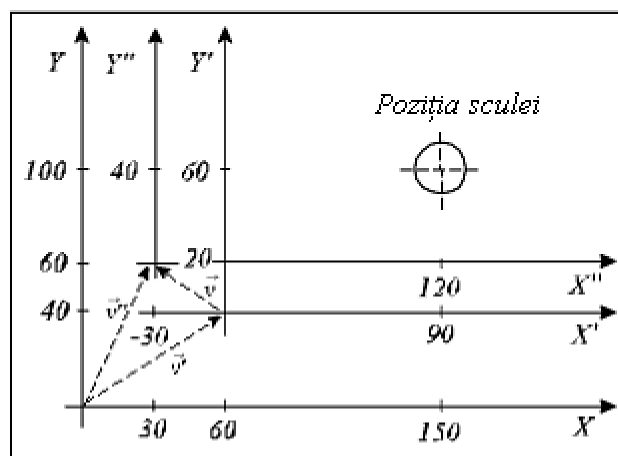


Fig. 9.3-1

Acum una dintre aceste două proceduri poate fi adoptată pentru a transfera sistemul de coordonate local în punctul de poziție X", Y".

- Cu o specificare de dată absolută: instrucțiunea (G 90) G 52 X30 Y60 va deplasa originea sistemului de coordonate X", Y" în punctul X=30, Y=60 în sistemul de coordonate X, Y. Componentele vectorului v" vor fi produse de către specificația v''_x=30, v''_y=60.
- Cu o specificare de dată incrementală: instrucțiunea G91 G52 X-30 Y60 va deplasa originea sistemului de coordonate X", Y" în punctul X= -30, Y=20 în sistemul de coordonate X', Y'. Componentele vectorului v" vor fi produse de către specificația v''_x=-30, v''_y=20. Indicarea locației noului sistem de coordonate local în sistemul de coordonate de lucru X, Z se face cu vectorul v"=v'+v. Componentele sale sunt: v''_x=60+(-30)=30, v''_y=40+20=60.

Poziția sculei în sistemul de coordonate X"," va fi X"=90, Z"=40.

Instrucțiunea

G90 G52 v0

va şterge ofsetul de la coordonatele specificate în v.

Sistemul de coordonate local va avea ofset în fiecare sistem de coordonate de lucru.

.

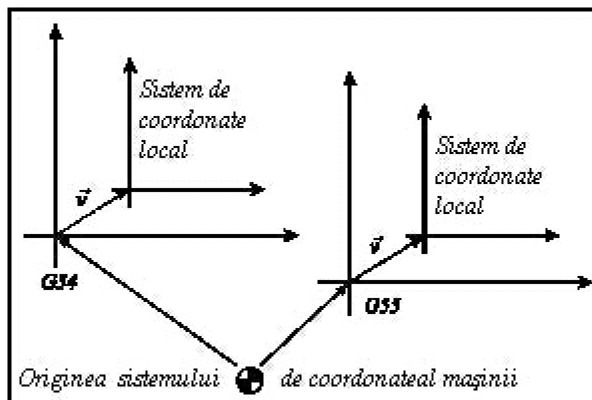


Fig. 9.3-2

Programarea instrucțiunii G92 va şterge ofseturile produse de către instrucțiunea G52 pe axele specificate în G92 – similar cu cazul în care s-ar da comanda G52 v0.

De câte ori scula se află în punctul de coordonate în X=200, Y=120 în sistemul de coordonate de lucru X, Y, instrucțiunea

```
G52 X60 Y40
```

îi va schimba poziția la X'=140, Y'=80 în sistemul de coordonate local X', Y'.
Acum instrucțiunea

```
G92 X110 Y40
```

va stabili poziția sculei la X"=110, Y"=40 în noul sistem de coordonate de lucru X", Y". Astfel sistemul de coordonate de lucru X', Y' va fi şters cu comanda G92 similar cum s-ar întâmpla dacă s-ar da comanda G52 X0 Y0

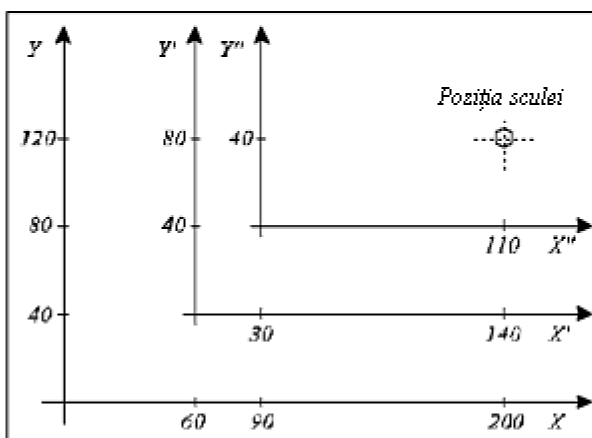


Fig. 9.3-3

Notă:
- ofsetul sistemului de coordonate local va fi şters prin executarea comenzilor M2, M30 şi/sau prin resetarea programului.

9.4 Selectarea planului (G17, G18, G19)

Planul în care se va executa
- interpolarea circulară,
- specificarea datelor cu coordonate polare,
- rotația sistemului de coordonate,
- compensarea razei sculei,
- poziționarea ciclurilor de găurire

se poate selecta cu următoarele coduri G:

G17 ………. planul X_p Y_p
G18 ………. planul Z_p X_p
G19 ………. planul Y_p Z_p

unde

X_p=X pe o axă paralelă la X,
Y_p=Y pe o axă paralelă la Y,

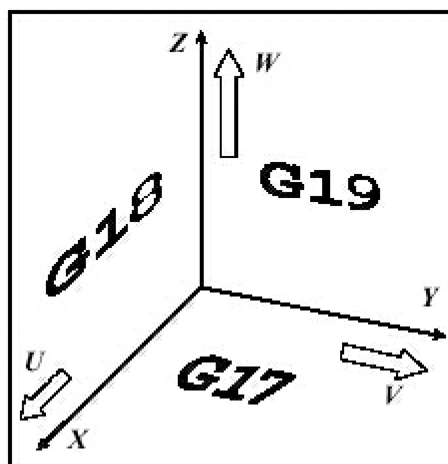


Fig. 9.4-1

Z_p=Z pe o axă paralelă la Z.
La planul selectat se face referire ca fiind „planul principal“.
Selectarea unei anumite axe paralele se va face (cu una dintre instrucțiunile G 17, G18 sau G19) funcție de adresa axei programate în blocul dat:
Când X şi U, Y şi V, Z şi W sunt axele paralele:

Planul XY va fi selectat cu G17 X_Y_,
Planul XV va fi selectat cu G17 X_V_,
Planul UV va fi selectat cu G17 U_V_,
Planul XW va fi selectat cu G18 X_W_,
Planul YZ va fi selectat cu G19 Y_Z_,
Planul VZ va fi selectat cu G19 V_Z_.

Dacă într-un bloc nu e specificată una din instrucțiunile G17, G18, G19, planul selectat rămâne neschimbat:

G17 X____Y____planul XY
U____Y____planul XY rămâne.

Dacă în blocul cu G17, G18, G19 nu e specificată o adresă de axă, comanda numerică va considera axele de bază:

planul XY va fi selectat cu G17,
planul XY va fi selectat cu G17 X,
planul UY va fi selectat cu G17 U,
planul XV va fi selectat cu G17 V,
planul ZX va fi selectat cu G18,
planul WX va fi selectat cu G18 W.

Comanda de mişcare nu afectează planul selectat:

```
(G90) G17 G00 Z100
```

va selecta planul XY, deplasând axa Z în punctul de coordonată 100.
După punerea sub tensiune, planul implicit (G17 sau G18) este specificat funcție de grupul de parametri *CODES*.
Planul principal poate fi selectat de mai multe ori în acelaşi program.
Adresele U, V, W pot fi selectate ca paralele, în parametri.

10 Funcția rotire arbore

10.1 Comanda vitezei arborelui (codul S)

Numărul maxim de cifre scrise la adresa **S** este 5 şi cu acest număr comanda numerică va trimite un cod la PLC.
Funcție de construcția maşinii, PLC-ul poate interpreta adresa S drept un cod sau ca un număr de rotații/minut.
Atunci când într-un anumit bloc se programează o comandă de deplasare şi o viteză de rotație a arborelui (S), funcția S se va aplica în timpul sau după comanda de mişcare.
Modul de execuție al comenzilor va fi definit de constructorul maşinii.
Vitezele specificate la adresele S sunt valori modale. La momentul punerii sub tensiune, comanda numerică va prelua valoarea S0. Viteza de rotație a arborelui are o valoare limită minimă şi una maximă în fiecare treaptă a cutiei de viteze. Acestea sunt definite de către constructorul maşinii în parametri şi comanda numerică nu permite viteze în afara acestei game.

10.2 Programarea controlului vitezei periferice constante

Funcția de control a vitezei periferice constante poate fi folosită doar în cazul unei acționări principale cu variație continuă. În acest caz, comanda numerică poate să modifice viteza de rotație a arborelui astfel ca viteza sculei să fie constantă relativ la suprafața piesei şi egală cu valoarea programată.

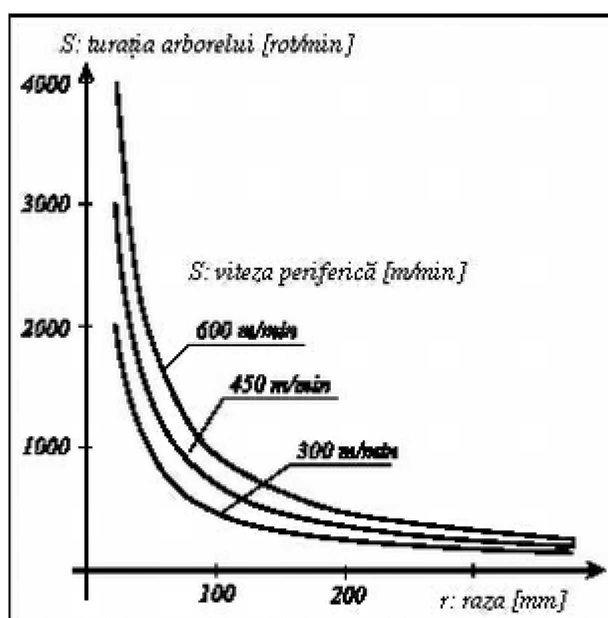


Fig. 10.2-1

Viteza periferică constantă trebuie să fie specificată în funcție de unitatea de intrare pe baza următorului tabel:

Unitatea de intrare	Unitatea de viteză periferică constantă
mm (G21 metric)	m/min
inci (G20 inci)	picior (feet) / min

10.2.1 Comanda controlului vitezei periferice constante (G96, G97)

Comanda

G96 S

activează funcția de control a vitezei periferice constante. Controlul vitezei periferice constante trebuie specificat la adresa S în unitatea de măsură dată în tabelul de mai sus.

Comanda

G97 S

anulează funcția de control a vitezei periferice constante. Viteza dorită a arborelui trebuie specificată la adresa S (în rotații/minut).

- Cu scopul calculului vitezei periferice constante sistemul de coordonate trebuie impus astfel ca punctul său de zero să coincidă cu axa de rotație.
- Controlul vitezei periferice constante este activ doar după ce se porneşte arborele cu ajutorul M3 sau M4.
- Valoarea este modală chiar şi după ce calcularea ei a fost anulată cu ajutorul comenzii G97. După punerea sub tensiune valoarea implicită a vitezei periferice constante este determinată de parametrul CTSURFSP.

```
G96 S100  (100 m/min sau 100 picioare/min)
G97 S1500 (1500 rot/min)
G96 X260  (100 m/min sau 100 picioare/min)
```

- Calculul vitezei periferice constante este de asemenea activ în starea G94 (avans/min).
- Dacă controlul vitezei periferice constante este anulat prin intermediul comenzii G97 şi nu se specifică o nouă viteză, rămâne activă ultima viteză pe care a avut-o arborele în starea G96.

```
G96 S100 (100 m/min sau 100 picioare/min)
   .
   .
   .
G97      (viteza de rotație corespunzătoare diametrului X)
```

- În cazul poziționării cu avans rapid (bloc G00), viteza periferică constantă nu este calculată continuu, dar se va calcula viteza de rotație corespunzătoare poziției finale. Aceasta pentru a se evita schimbări inutile ale vitezei de rotație a arborelui.
- Pentru a se calcula viteza periferică constantă, trebuie impus la axa de rotație a arborelui punctul zero al axei, presupunând că viteza de rotație a arborelui se modifică.

10.2.2 Limitarea vitezei periferice constante (G92)

Prin intermediul comenzii

G92 S

se poate impune cea mai mare viteză de rotație a arborelui în cazul în care e activ controlul vitezei periferice constante. În timpul calculului vitezei periferice constante, comanda numerică limitează viteza de rotație a arborelui la această valoare. În acest caz unitatea pentru S este rot/min.

- După punerea sub tensiune, la fel ca în cazul în care valoarea lui S nu a fost limitată cu comanda G92, limita superioară a vitezei de rotație a arborelui în cazul controlului vitezei periferice constante este valoarea maximă permisă pentru acea treaptă a cutiei de viteze.
- Valoarea maximă a vitezei de rotație a arborelui este modală până când este programată o nouă valoare sau până ce se opreşte comanda numerică.

10.2.3 Selectarea unei axe pentru controlul vitezei periferice constante

Axa în funcție de a cărei poziție se calculează viteza periferică constantă este selectată cu parametrul 1182 AXIS. Numărul axei logice trebuie înscris la parametru.
Dacă trebuie folosită o altă axă decât cea selectată, se poate specifica axa funcție de care trebuie calculată viteza periferică constantă cu ajutorul comenzii:

G96 P

Interpretarea pentru adresa P este:

P1: X, P2: Y, P3: Z,
P4: U, P5: V, P6: W,
P7: A, P8: B, P9: C

- Valoare impusă la adresa P este modală. După punerea sub tensiune se va activa ca axă funcție de a cărei poziție se calculează viteza periferică constantă, axa impusă la parametrul AXIS.

10.3 Reacția de poziție a arborelui

În prelucrarea normală comanda numerică va produce o comandă a vitezei pentru variatorul de turație al arborelui, proporțională cu viteza programată (valoarea specificată la adresa S). Acum, acest variator va lucra în modul de control al vitezei.
Totuşi, anumite procese tehnologice necesită ca arborele să fie adus într-o anumită poziție unghiulară. Aceasta se numeşte „poziționarea arborelui“ sau „indexare“.
Anterior poziționării, comanda numerică va trece variatorul arborelui în modul de control al poziției. În practică aceasta înseamnă că în locul unui comenzi a vitezei proporționale cu codul S, comanda numerică va măsura poziția arborelui cu ajutorul unui codor de poziție montat pe arbore şi va produce o comandă către variator în concordanță cu deplasarea unghiulară dorită (similar cu restul axelor controlate). Aceasta este reacția de poziție.
Pentru a fi posibilă poziționarea arborelui, pe o anumită maşină, este necesar ca să existe un codor de poziție montat pe arbore şi ca variatorul arborelui să fie capabil să opereze şi în modul cu reacție de poziție.

10.4 Oprirea orientată a arborelui

„Orientarea arborelui“ şi „oprirea orientată a arborelui“ fac referire la funcționarea cu oprirea arborelui într-o anumită poziție unghiulară. Aceasta poate fi necesar de exemplu pentru o schimbare automată a sculei sau pentru executarea anumitor cicluri de găurire. Posibilitatea orientării pe o anumită maşină trebuie specificată cu parametrul *ORIENTI,* în parametri.
Comanda orientării arborelui este activată cu funcția M19, dar se poate produce şi de către alte funcții depinzând de respectiva maşină unealtă. Orientarea poate fi executată într-unul din următoarele două moduri :

- Dacă arborele nu se poate folosi în modul de control al poziției, orientarea se poate face rotind arborele față de un senzor de proximitate montat pe maşină.
- Dacă arborele se poate folosi în modul de control al poziției, la comanda M19 comanda numerică va aduce arborele în poziția pulsului de zero al codorului de poziție. Comanda numerică va închide automat bucla de control a poziției.

10.5 Poziționarea arborelui (indexarea)

O poziționare a arborelui se poate face numai după ce bucla de control a poziției arborelui a fost închisă după orientare. Prin urmare, această funcție este folosită pentru închiderea buclei. Bucla va fi deschisă de comenzile de rotire M3 sau M4.
Dacă valoarea parametrului *INDEX*=1 (indicând că bucla de control a poziției pentru acționarea principală poate fi închisă) şi valoarea parametrului *INDEX_C1*=0, se va executa indexarea arborelui cu funcția M.
În aceste condiții funcția M de prag şi parametrii de la *M_NUMB1* la *M_NUMB1*+360 vor fi interpretate ca o comandă de indexare a arborelui, adică numărul de prag va fi scăzut din valoarea programată M, şi numărul obținut va fi tratat ca fiind mărimea incrementului specificată în grade.
Astfel, dacă *M_NUMB1*=100, comanda M160 înseamnă că arborele trebuie să fie rotit cu 160-100=60 de grade din poziția actuală. Direcția în care se face rotația este selectată cu parametrul *CDIRS1*, iar viteza sa este selectată cu parametrul *RAPIDS1*.

10.6 Detectarea fluctuației vitezei arborelui (G25, G26)

Comanda

G26

activează detectarea fluctuației vitezei arborelui, în timp ce comanda

G25

o anulează. După punerea sub tensiune sau RESET comanda numerică are activată starea G26, adică detectarea fluctuației vitezei arborelui este activă. Această funcție semnalizează anomaliile care apar în decursul rotirii arborelui, putând evita astfel de exemplu griparea arborelui.
Detectarea fluctuației vitezei arborelui este influențată de 4 parametri. Aceşti parametri pot fi rescrişi dintr-un program cu adrese succesive comenzii G26. La deconectarea alimentării, parametrii rescrişi sunt menținuți. Parametrii sunt rescrişi ca efect al comenzii

G26 Pp Qq Rr Dd.

Tabelul următor conține interpretările parametrilor:

nume	parametru	semnificație	unitate	valoare limită
p	5001 TIME	timpul care a trecut de la lansarea unei noi comenzi pentru viteza arborelui până la începerea verificării	100 ms	65535
q	5002 SCERR	toleranța vitezei specificate a arborelui	%	1-50
r	5003 FLUCT%	mărimea permisă pentru fluctuația vitezei arborelui în procente din viteza programată	%	1-50
d	5004 FLUCTW	fluctuația vitezei arborelui în valoare absolută	rot/min	65535

Procesul de detectare a fluctuației vitezei arborelui decurge după cum urmează.

Pornirea detectării fluctuației vitezei arborelui

Ca efect al unei noi viteze de rotație detectarea este suspendată de către comanda numerică. Detectarea fluctuației vitezei arborelui porneşte când:

- viteza curentă a arborelui atinge valoarea specificată în limita de toleranță determinată de valoarea „q“, sau

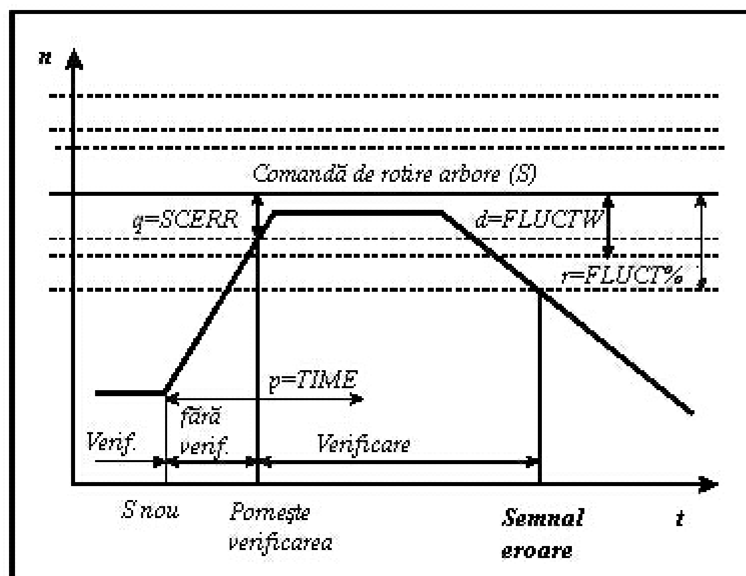


Fig. 10.6-1

- viteza curentă a arborelui nu a atins valoarea specificată în limita de toleranță determinată de valoarea „q“, dar a trecut timpul determinat de valoarea „p“ din momentul comenzii.

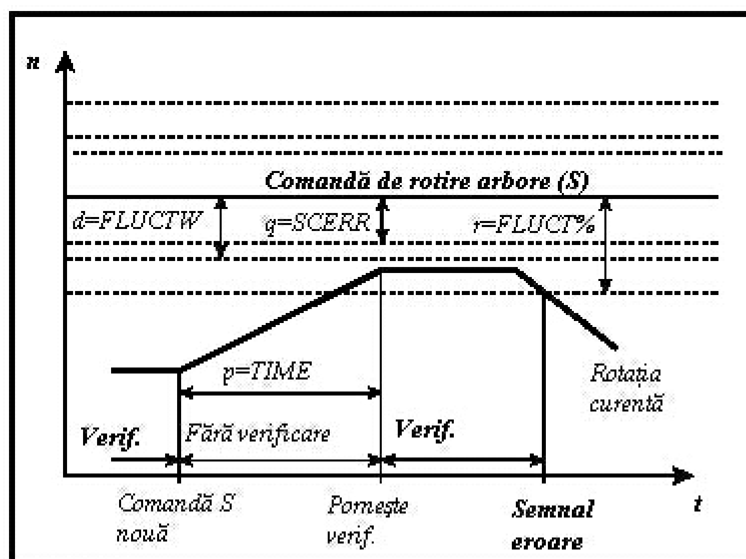


Fig. 10.6-2

Detectarea erorii

În cursul detectării, comanda numerică trimite un mesaj de eroare în cazul în care diferenţa între vitezele de rotaţie curentă şi specificată depăşeşte:

- limita de toleranţă specificată de valoarea „r“ în procente din valoarea comandată şi
- de asemenea limita de toleranţă absolută specificată de valoarea „d“.

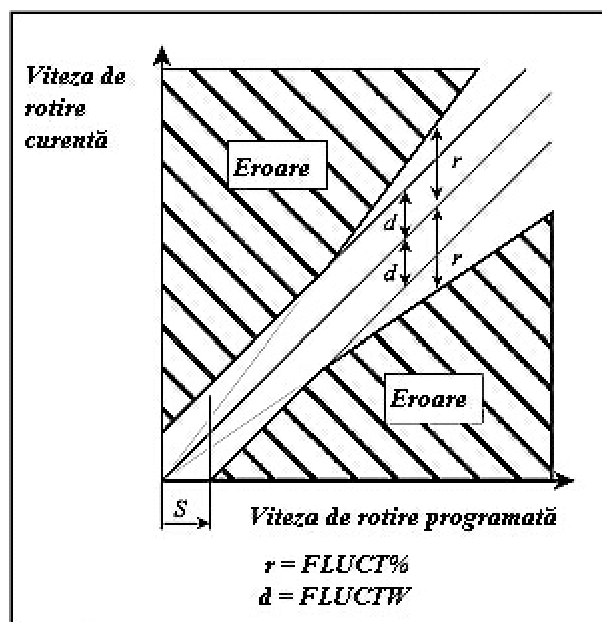


Fig. 10.6-3

Atunci când viteza curentă depăşeşte ambele limite de toleranţă, comanda numerică impune flag-ul I656 la PLC. Gama de viteză în care comanda numerică produce alarma se poate observa în figura alăturată. Dacă viteza arborelui specificată este sub valoarea „S“ care apare în figură, comanda numerică produce alarma, considerând că viteza curentă este de 0 rot/min pentru mai mult de 1 secundă.

- Detectarea fluctuaţiei vitezei arborelui este activă doar dacă pe arbore este montat un codor.
- Viteza specificată a arborelui, funcţie de care se face detectarea este calculată luînd in consideraţie următoarele: corecţia, limitele gamei de turaţie şi turaţia maximă programată (G92 S_) la calculul vitezei periferice constante (G96).
- Detectarea fluctuaţiei vitezei arborelui este activă doar în cazul G26 şi al rotirii arborelui (starea M3 sau M4).
- Comanda G26 trebuie programată într-un bloc singular.

11 Funcția schimbare sculă

11.1 Comanda de selectare sculă (codul T)

La adresa **T** pot fi înscrise un număr de maxim patru cifre din a căror interpretare comanda numerică va transmite un cod către PLC. Când într-un bloc sunt programate o comandă de mişcare şi un număr de sculă (T), funcția T va fi executată în timpul sau după executarea comenzii de mişcare. Metoda de executare este determinată de constructorul maşinii.

11.2 Formatul de program pentru programarea numărului sculei

Există două moduri principale de a se face referire la o schimbare de sculă în cursul programului piesă. Acestea depind de configurația maşinii. Tehnica particulară de apelare a sculei (aplicabilă în programul piesă) este definită de către constructorul maşinii.

Cazul A

O schimbare de sculă poate fi realizată pe maşină manual ori cu ajutorul unui schimbător de scule de tip revolver. Când se face referire la codul T:
- în cazul schimbării manuale a sculei, numărul sculei cerute apare pe afişaj; scula trebuie să fie prinsă în arbore manual. În continuare prelucrarea va fi reluată după ce se dă start.
- în cazul unui schimbător de scule de tip revolver, ca urmare a acțiunii codului T scula cerută va fi adusă în poziția de prelucrare în mod automat.

Astfel o referire la un număr de sculă va determina o schimbare imediată în blocul în care a fost specificat T.

Cazul B

O schimbare de sculă necesită anumite pregătiri ale maşinii. Sunt necesare următoarele etape:
- Scula cerută trebuie să fie găsită în magazia de scule. Referirea făcută la adresa T în programul piesă va aduce scula cerută în poziția de schimbare. Această operație este executată în timp ce maşina execută operația anterioară de prelucrare a piesei.
- Axele (sau numai unele dintre ele) trebuie trimise în poziția de schimbare a sculei.
- Schimbarea sculei este executată prin intermediul funcției M06 din program. Comanda numerică va aştepta executarea schimbării sculei până când scula T (cerută) este adusă în poziția de schimbare. Prin executarea comenzii, scula nouă va fi plasată pe arbore. În continuare se poate relua prelucrarea.
- Scula folosită anterior este reintrodusă în magazie. Această operație este executată în timp ce maşina execută prelucrarea piesei.
- Se declanşează căutarea în magazie a sculei cu care se va face următoarea prelucrare.

Această procedură este descrisă în programul piesă după cum urmează:

Program piesă	Explicații
…………	
….Tnnnn………	căutarea sculei Tnnnn
…………	continuă executarea programului piesă, căutarea sculei se execută în fundal
…M06 Tmmmm…	scula T*nnnn* este plasată pe arbore,
…………	scula folosită anterior este reintrodusă în magazie
…………	se porneşte căutarea sculei T*mmmm*, în timp ce maşina execută prelucrare
…M06 Tpppp….	scula T*mmmm* este plasată pe arbore
…………	scula T*nnnn* este reintrodusă în magazie, se porneşte căutarea sculei T*pppp*
…………	în acest timp se execută prelucrare

12 Funcții diverse (M) şi auxiliare

12.1 Funcții diverse (coduri M)

Codul **M** având specificată la adresă o valoare numerică de maxim 3 cifre, este transferat de către comanda numerică la PLC.
Atunci când într-un bloc sunt programate o comandă de mişcare şi o funcție M, funcția M va fi executată în timpul sau după executarea comenzii de mişcare. Modul de executare este definit de constructorul maşinii.
Codurile M includ funcții standard, care pot fi folosite doar pentru funcții selectate special. Acestea sunt:

M00, M01, M02, M30, M96, M97, M98, M99: coduri de control program
M03, M04, M05, M19: coduri rotire arbore
M06: cod de schimbare sculă
M07, M08, M09: coduri utilizare răcire
M11, …, M18: coduri schimbare treaptă de turație arbore.

Restul valorilor M pot fi folosite fără restricții.
Atunci când indexarea este declanşată de o funcție M, codurile M ale indexării arborelui sunt selectate pe baza unui parametru.
Comanda numerică permite ca să fie scrise într-un anumit bloc un număr de coduri M din grupe diferite. În acest caz totuşi, codurile M au o secvență prestabilită de execuție. Grupele şi secvența de execuție sunt:

Grupa 1	M06 (schimbare sculă)
Grupa 2	M11, …, M18 (schimbare treaptă de turație arbore)
Grupa 3	M03, M04, M05, M19 (rotire arbore)
Grupa 4	M07, M08, M09 (utilizare răcire)
Grupa 5	M*nnn* (oricare altă funcție M)
Grupa 6	coduri M de indexare arbore
Grupa 7	M00, M01, M02, M30, M96, M97, M98, M99 (coduri de control program)

Numărul de funcții M care poate fi programat într-un anumit bloc este de 5. Într-un bloc se poate comanda doar un singur cod M din fiecare grupă. Programarea conflictuală va produce mesajul *3032 CONFLICTING M CODES (coduri M conflictuale).*
Modul de funcționare exact al fiecărui cod M este definit de către constructorul maşinii respective pentru a satisface cerințele. Singura excepție sunt codurile de control program.

<u>Codurile de control program sunt:</u>

M00= oprire programată

Condiția de oprire va fi generată la sfârşitul blocului în care a fost specificat M00. Toate funcțiile modale rămân neschimbate. Se poate reporni cu START.

M01= stop condiționat

Efectul ei este identic cu al codului M00. Va fi executat când este activată tasta **STOP CONDIȚIONAT** . Nu are efect dacă nu se apasă tasta corespunzătoare.

M02, M30= sfârşit de program

Înseamnă sfârşitul programului principal. Sunt oprite toate operațiile şi comanda numerică este resetată. Maşina va fi resetată de către programul PLC. Funcție de parametrul *PRTCNTM*, fiecare execuție a comenzii M02 sau M30 poate incrementa numărătorul de piese prelucrate.

M98= apelare subprogram (subrutină)
Va apela un subprogram (subrutină).
M99= sfârşit subprogram (subrutină)
Va permite ca execuția programului să revină în locul de unde s-a făcut apelarea.

12.2 Funcții auxiliare (coduri A, B, C)

La fiecare din adresele **A, B, C** pot fi specificate maximum trei cifre dacă una dintre ele (sau toate) este selectată ca funcție auxiliară în parametri. Valoarea specificată pentru funcția auxiliară va fi transferată către PLC.
Atunci când sunt programate într-un bloc o comandă de mişcare şi o funcție auxiliară, funcția A, B, C va fi executată pe timpul sau după executarea comenzii de mişcare. Modul de execuție va fi definit de constructorul maşinii.
De exemplu indexarea mesei se poate face la adresa B.

12.3 Succesiunea de executare a funcțiilor M

Funcțiile M scrise într-un anumit bloc vor fi executate de către comanda numerică în următoarea succesiune:

1.	Schimbare sculă:	M06
2.	Apel sculă:	T
3.	Selectare treaptă turație:	M11, …,M18
4.	Turație arbore:	S
5.	Control arbore:	M03, M04, M05, M19
6.	Răcire:	M07, M08, M09
7.	Alte funcții M:	M*nnn*
8.	Indexare arbore:	Cu funcții M
9.	Funcția A:	A
10.	Funcția B:	B
11.	Funcția C:	C
12.	Coduri de control program:	M00, M01, M02, M30, M96, M97, M98, M99

Dacă succesiunea de mai sus nu satisface cerințele, blocul trebuie împărțit în alte câteva blocuri, cu funcțiile M scrise în succesiunea dorită în fiecare bloc.

13 Configurarea programului piesă

Structura programului piesă a fost deja descrisă în introducere, prezentându-se codurile şi formatele programelor în memorie. Această secțiune va discuta procedurile de organizare a programelor piesă.

13.1 Număr de secvență (adresă N)

Blocurile programului pot fi specificate cu numere de serie sau secvență. Numerotarea se poate face la adresa **N**. Blocurile pot fi numerotate cu maximum 5 cifre la adresa N. Folosirea adresei N nu este obligatorie. Anumite blocuri pot fi numerotate, altele nu. Numerele de bloc nu trebuie să fie în ordine consecutivă.

13.2 Salt de bloc condiționat

Un salt de bloc condiționat poate fi programat la adresa barată / . Valoarea adresei barate poate fi de la 1 la 9. Cifrele de la 1 la 9 reprezintă numărul de comutatoare.
Comutatorul **BLOC CONDIȚIONAT** nr.1 poate fi găsit pe panoul operatorului al comenzii numerice. Celelalte comutatoare pot fi prevăzute opțional, semnalele lor putând fi introduse prin interfața comenzii numerice.
Dacă un salt de bloc condiționat /n este programat la începutul unui bloc,
- acel bloc va fi omis din execuție atunci când al n-lea comutator este on,
- acel bloc va fi executat când al n-lea comutator este off.

13.3 Program principal şi subprogram

Sunt diferențiate două programe diferite – program principal şi subprogram. În prelucrarea unei piese pot fi implicate activități repetitive, care pot fi descrise cu un detaliu particular de program. Pentru a evita scrierea repetitivă a detaliului de program, acesta poate fi organizat într-un subprogram care să fie apelat din programul piesă. Structura programului principal şi cea a subprogramului au fost descrise în Introducere.
Diferența dintre ele este aceea că, în timp ce prelucrarea este terminată după executarea programului principal, comanda numerică fiind în aşteptarea unui alt START, după executarea unui subprogram, execuția revine în programul apelant, reluând procesul de prelucrare din acel punct.
În termeni de tehnică de programare, diferența dintre cele două programe stă în modul de terminare a programului. Sfârşitul programului principal este specificat cu codurile M02 sau M30 (nu sunt obligatorii), în timp ce subprogramul trebuie terminat cu codul M99.

13.3.1 Apelarea subprogramului

Seria de instrucțiuni

M98 P…

va genera un apel de subprogram. Ca rezultat, execuția programului va continua în subprogramul al cărui număr este definit la adresa P. Limitele pentru adresa P sunt 1 şi 9999. După executarea subprogramului, prelucrarea va fi continuată în programul principal cu blocul următor celui în care s-a apelat subprogramul.

```
program principal   subprogram      comentariu

O0010                                     execuţia programului
                                          (principal) O0010
.....
.....
M98 P0011         ---->       O0011       apelul subprogramului O0011
                              .....
                              .....
                              .....
blocul următor    <----       M99         revenirea în programul
                                           apelant
....                                      reluarea programului O0010
....
```

Seria de instrucţiuni

M98 P…. L….

va apela subprogramul (specificat la adresa P) în mod repetat în succesiunea specificată la adresa L. Limitele pentru adresa L sunt 1 şi 9999. Dacă nu se dă o valoare pentru L, subprogramul va fi apelat o singură dată, comanda numerică interpretând L=1.
Instrucţiunea M98 P11 L6 semnifică faptul că subprogramul 011 trebuie apelat în mod repetat de şase ori.
Este posibil de asemenea să se apeleze un subprogram din alt subprogram. Apelurile de subprograme din subprograme pot merge până la nivelul 4.

```
program principal      subprogram    subprogram    subprogram    subprogram
O0001           ------>O0011     --->O0012     --->O0013      ---> O0014
……              |       ….       |     ….       |     ……       |     ………
……              |       ….       |     …..      |     ……       |     ……..
M98P11_____|           M98P12__|     M98P13__|     M98P14__|         …….
…….   <-----           ……  <---      ……  <---      …….  <---         …….
…….             |      ……        |    ……        |    …….        |     …….
M02             |_____ M99       |__ M99        |___M99        |___ M99
```

Note:

- dacă este depăşit numărul de patru apeluri de subprograme din subprograme, este afişat mesajul de eroare *3069 LEVEL EXCESS (depăşire nivel).*
- dacă este depăşită valoarea de 9999 pentru adresa P sau aceasta lipseşte, este afişat mesajul de eroare *3017 MISSING OR FAULTY P (P lipsă sau greşit).*
- dacă valoarea pentru L este incorectă, este afişat mesajul de eroare *3072 DEFINITION ERROR L (eroare de definire L).*
- dacă este specificat cu un identificator la adresa P un program inexistent în memorie, este afişat mesajul de eroare *3073 NOT EXISTING PROGRAM (program inexistent).*

13.3.2 Revenirea dintr-un subprogram

Utilizarea instrucţiunii

M99

într-un subprogram semnifică sfârşitul acelui subprogram şi revenirea execuţiei în programul principal, la blocul următor celui din care s-a făcut apelul.

```
program principal      subprogram   comentariu

O0010                               executarea programului O0010
....
....
....
N101 M98 P0011    ----> O0011       apelarea subprogramului O0011
                        ....        executarea subprogramului O0011
                        ....
                        ....
N102 ....         <---- M99         revenirea la blocul următor al
                                    programului apelant
....                                reluarea programului O0010
....
```

Utilizarea instrucțiunii

M99 P…

într-un subprogram înseamnă sfârşitul acelui subprogram şi execuția programului revine la blocul programului apelant specificat la adresa P. În acest caz valoarea limită pentru P este 99999.

```
program principal       subprogram  comentariu

O0010                               executarea programului O0010
...
...
...
N101 M98 P0011    ----> O0011       apelarea subprogramului O0011
                        ....        executarea subprogramului O0011
                        ....
                        ....
N250 ...          <---- M99 P250    revenirea la blocul N250 al
                                    programului apelant
...                                 reluarea programului O0010
...
```

Instrucțiunea

M99 (P….) L….

va reînscrie numărătorul de cicluri al programului apelant. Dacă se scrie pentru L valoarea 0, subprogramul va fi apelat numai o dată. Dacă, de exemplu, subprogramul O0011 este apelat cu instrucțiunea M98 P11 L20 şi se face o revenire cu instrucțiunea M99 L5, subprogramul O0011 va fi apelat de 6 ori. (Valorile limită pentru L sunt de la 1 la 9999.)

Notă:

- Se afişează un mesaj de eroare *3070 NOT EXISTING BLOCK NO. P (nu există blocul nr. P)* când numărul blocului de revenire (P) nu este găsit în programul apelant.

13.3.3 Salt în interiorul programului principal

Folosirea instrucțiunii

M99

în programul principal va produce un salt necondiționat la blocul specificat la adresa P a programul principal şi execuția programului va fi continuată de acolo. Folosirea acestei instrucțiuni poate produce un ciclu fără sfârşit:

```
O0123
N1... <----
...        |
...        |
...        |
M99  ______|
```

Folosirea instrucțiunii

M99 P….

va produce un salt necondiționat la blocul specificat la adresa P a programului principal şi execuția programului va fi continuată de acolo. Folosirea acestei instrucțiuni poate produce un ciclu fără sfârşit:

```
O0011                          O0011
...                            ...
...                            M99 P225 __
N128... <----                  ...        |
...          |                 ...        |
...          |                 N225 <-----
M99 P128 ____|                 ...
```

Posibilitatea apariției ciclurilor fără sfârşit poate fi evitată prin specificarea blocului conținând instrucțiunea M99 în forma /1 M99. Acum saltul va fi omis sau nu, funcție de starea comutatorului de bloc condiționat.

14 Compensarea sculei

14.1 Referirea la valorile de compensare a sculei (H şi D)

Se poate face referire la
compensarea lungimii sculei la adresa **H**,
compensarea razei sculei la adresa **D**.
Numărul care urmează adresa (numărul de compensare a sculei) indică valoarea de compensare particulară care să fie aplicată. Valorile limită ale adreselor H şi D sunt cuprinse între 0 şi 999.
Tabelul următor prezintă împărțirea memoriei de compensare:

	Cod H		**Cod D**	
număr compensare	**geometrie**	**uzură**	**geometrie**	**uzură**
01	-350.200	0.130	-32.120	0.012
02	830.500	-0102	52.328	-0.008
.		.		.
.		.		.
.		.		.

Numărul compensării 00 nu este inclus în acest tabel, valorile compensărilor fiind în acest caz întotdeauna zero.
Valoarea geometriei – lungimea respectiv raza sculei. Este un număr cu semn.
Uzura – uzura sculei care apare în decursul prelucrării. Este un număr cu semn.
De câte ori se face referire în program la o compensare la adresele H sau D , comanda numerică va lua în calcul pentru compensare suma dintre valoarea geometriei şi uzură. Dacă de exemplu, se face referire în program la H2, mărimea compensării va fi conform tabelului de mai sus 830.500+(-0.102)=830.398.
Adresele H şi D sunt modale, deci comanda numerică va ține cont o anumită valoare pentru compensare până când se dă o altă comandă pentru D sau H. Cu alte cuvinte, după ce s-a citit valoarea compensării cu comanda D sau H, valoarea respectivă nu va fi afectată de o modificare în tabelul valorilor de compensare (de exemplu prin programarea G10).
Valorile de compensare se vor păstra în memorie şi după deconectarea de la rețea.
Memoria de compensare poate fi salvată în memorie ca un program piesă.

Mărimea limitelor pentru valorile ofsetului pentru geometrie şi pentru uzură sunt:

unităţi de intrare	unităţi de ieşire	sistem increment	valoare geometrie	valoare uzură	unitate de măsură
mm	mm	IR-A	±0.01÷99999.99	±0.01÷163.80	mm
		IR-B	±0.001÷9999.999	±0.001÷16.380	
		IR-C	±0.0001÷999.9999	±0.0001÷1.6380	
inci	mm	IR-A	±0.001÷9999.999	±0.001÷6.448	inci
		IR-B	±0.0001÷999.9999	±0.0001÷0.6448	
		IR-C	±0.00001÷99.99999	±0.00001÷0.06448	
inci	inci	IR-A	±0.001÷9999.999	±0.001÷16.380	inci
		IR-B	±0.0001÷999.9999	±0.0001÷1.6380	
		IR-C	±0.00001÷99.99999	±0.00001÷0.16380	
mm	inci	IR-A	±0.01÷99999.99	±0.01÷416.05	mm
		IR-B	±0.001÷9999.999	±0.001÷41.605	
		IR-C	±0.0001÷999.9999	±0.0001÷4.1605	

Compensările sculei pot fi selectate şi/sau modificate fie de la panoul operatorului din ecranul OFFSET fie prin program folosind instrucţiunea G10. Dacă se modifică compensarea curentă cu comanda G10, trebuie să se facă din nou referire la registrul de compensare curent D sau H, altfel valoarea modificată nu va fi luată în considerare. Valorile limită ale adreselor H sau D pentru un anumit sistem de control, adică numarul compensărilor de lungime sau rază care să fie specificate în comanda numerică, sunt determinate de către configuraţia memoriei comenzii numerice. În cazul unei configuraţii de memorie minime, numărul de compensări este 99, adică valorile limită ale adreselor H şi D sunt între 0 şi 99.

14.2 Modificarea valorilor de compensare a sculei din program (G10)

Instrucţiunea:

G10 R L P

poate fi folosită pentru modificarea compensărilor sculei din program. G10 este o instrucţiune cu o singură acţiune. Adresele şi valorile lor au următoarele semnificaţii:
Valoarea compensării este specificată la adresa R. În cazul G90 (comandă de specificare de dată absolută), valoarea introdusă la adresa R va fi transferată registrului de compensare corespunzător. În cazul G91 (comandă de specificare de dată incrementală) sau când este aplicat operatorul I, data înscrisă la adresa R va fi adunată la conţinutul registrului de compensare corespunzător.
Valoarea compensării care să fie modificată este specificată la adresa L:

L=10: se aplică valorii geometriei la compensarea lungimii (cod H),
L=11: se aplică uzurii la compensarea lungimii (cod H),
L=12: se aplică valorii geometriei la compensarea razei (cod H)
L=13: se aplică uzurii la compensarea razei (cod H).

Numărul valorii compensării care să fie modificată este precizat la adresa P.

Notă: Pentru o modificare programată a compensării razei, valoarea specificată la adresa R trebuie să fie interpretată ca o rază, în fiecare caz, indiferent de starea parametrului *TOOLRAD* . Comanda numerică va afişa mesajul *3001 VALUE EXCESS X, Y, ...F (depăşire valoare X, Y, ...F)* ori de câte ori valoarea specificată depăşeşte limitele indicate în tabelul anterior.

14.3 Compensarea lungimii sculei (G43, G44, G49)

Instrucţiunea

G43 *q* H sau

G44 *q* H

va impune modul de compensare a lungimii sculei.
Adresa *q* desemnează axa *q* la care se aplică compensarea lungimii sculei (*q* = X, Y, Z, U, V, W, A, B, C).
Adresa H înseamnă celula de compensare din care se ia valoarea compensării lungimii sculei.
Independent de faptul că *q* este o dată absolută sau incrementală, instrucţiunea G43 va aduna valoarea de compensare (specificată la adresa H) la coordonata punctului final obţinută în decursul execuţiei:

G43: + compensarea

Independent de faptul că *q* este o dată absolută sau incrementală, instrucţiunea G44 va scădea valoarea de compensare (specificată la adresa H) din coordonata punctului final obţinută în decursul execuţiei:

G44: - compensarea

Deoarece a fost programat incremental Z0, fiecare din instrucţiunile G43 G91 Z0 H1 şi G44 G91 Z0 H1 va produce deplasament egal cu lungimea sculei. Pentru G43, deplasamentul va fi pozitiv sau negativ în funcţie de semnul valoarea compensării de la H1. Cazul este exact invers pentru G44. După ce s-a executat comanda, poziţia afişată la coordonata Z va fi aceeaşi cu cea anterioară deoarece poziţia vârfului sculei va fi afişată după impunerea compensării de lungime.
Compensările sculei pot fi definite pe câteva axe simultan. De exemplu:

```
G43 Z250 H15
G43 W310 H16
```

Când într-un bloc sunt selectate câteva axe, compensarea lungimii sculei va fi luată în consideraţie pentru fiecare axă selectată:

```
G44 X120 Z250 H27
```

Când valoarea de compunere este modificată prin apelarea unei noi adrese H, valoarea anterioară va fi ştearsă şi noua valoare va deveni efectivă:

```
H1=10, H2=20
G90 G00
G43 Z100 H1.......deplasare la Z=110
G43 Z100 H2.......deplasare la Z=120
```

Efectele instrucţiunilor G43 şi G44 sunt modale până când este primită o nouă comandă din acel grup.

Comanda

G49 sau

H00

va anula compensarea lungimii sculei pe fiecare axă – cu deplasare dacă în bloc a fost

programată o comandă de mişcare sau cu transformare dacă nu este comandă de mişcare. Diferența dintre cele două comenzi este că H00 va şterge doar compensarea lăsând stările G43 sau G44 neafectate. Dacă în continuare se va face referire la o adresă H alta decât 0, se va impune noua compensare de lungime a sculei în funcție de starea G43 sau G44.

Dacă este folosită instrucțiunea G49, orice referire la adresa H nu va avea efect până când nu se va programa G43 sau G44.

La punerea sub tensiune, valoarea definită în grupul de parametri *CODES* decide care cod va fi activ (G43, G44, G49).

Exemplul de mai jos prezintă o operație simplă de găurire în care se ține cont de compensarea lungimii sculei:

lungimea sculei de găurire, H1=400

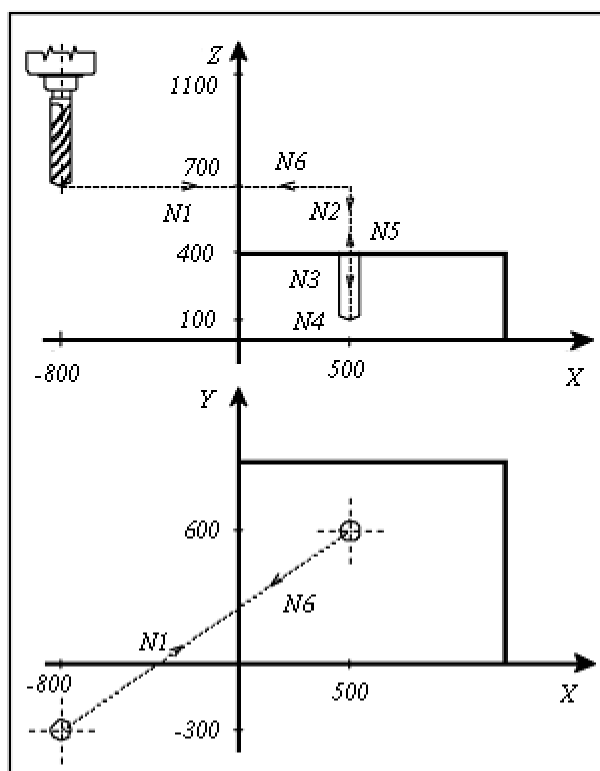


Fig. 14.3-1

```
N1 G90 G0 X500 Y600      (poziționare în planul X, Y)
N2 G43 Z410 H1           (deplasare la Z410 cu H1 compensare de lungime)
N3 G1 Z100 F180          (găurire până la Z100 cu avans F180)
N4 G4 P2                 (întârziere pentru 2 secunde)
N5 G0 Z1100 H0           (îndepărtarea sculei şi anularea compensării
                         lungimii; vârful sculei este în punctul x700)
N6 X-800 Y-300           (revenirea cu avans rapid în planul X, Y)
```

14.4 Ofsetul sculei (G54…G48)

G45 creşte mărimea deplasării cu valoarea ofsetului
G46 descreşte mărimea deplasării cu valoarea ofsetului
G47 creşte mărimea deplasării cu dublul valorii ofsetului
G48 descreşte mărimea deplasării cu dublul valorii ofsetului

Oricare dintre comenzile G45…G48 va fi activă împreună cu compensarea selectată cu codul D până când este apelată o altă valoare în legătură cu o comandă G45.. G48.

Fiind coduri nemodale, acestea sunt active doar în blocul în care au fost specificate.

În cazul unei specificări de dată absolută, mărimea deplasării va fi diferența dintre punctul final definit în blocul curent şi punctul final definit în blocul anterior. Orice creştere sau descreştere se referă la direcția mişcării produse în acest fel.

Cu programarea G45 (creştere cu valoarea ofsetului)

a. comandă de deplasare: 20
compensare: 5

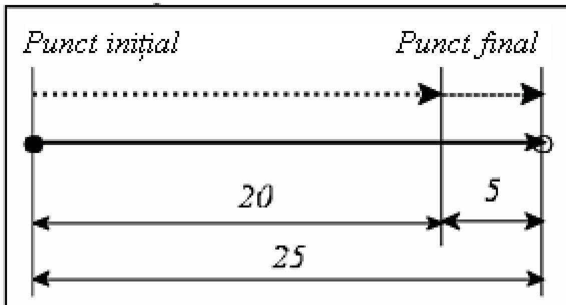


Fig. 14.4-1

b. comandă de deplasare: 20
compensare: -5

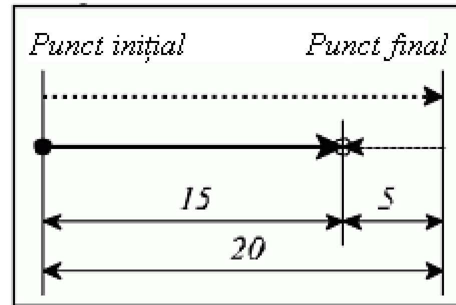


Fig. 14.4-2

a. comandă de deplasare: -20
compensare: 5

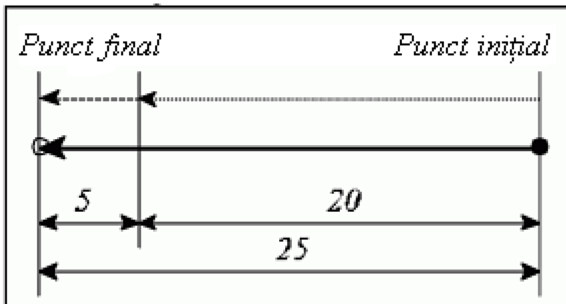


Fig. 14.4-3

b. comandă de deplasare: -20
compensare: -5

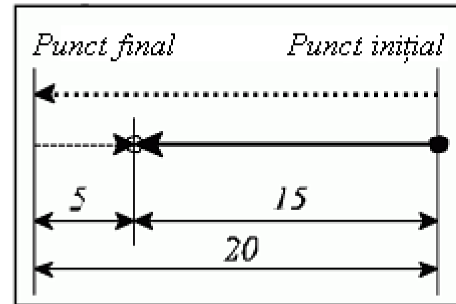


Fig. 14.4-4

Cu programarea G46 (descreştere cu valoarea ofsetului)

a. comandă de deplasare: 20
compensare: 5

cazurile b, c, d sunt similare cu cele de la G45

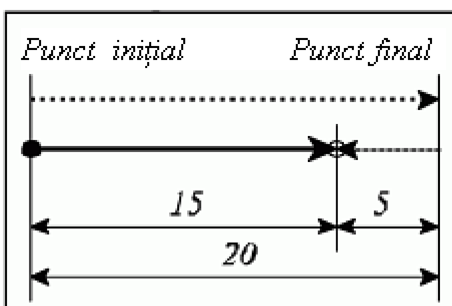


Fig. 14.4-5

Cu programarea G47 (creştere cu dublul valorii ofsetului)

a. comandă de deplasare: 20
compensare: 5

cazurile b, c, d sunt similare cu cele de la G45

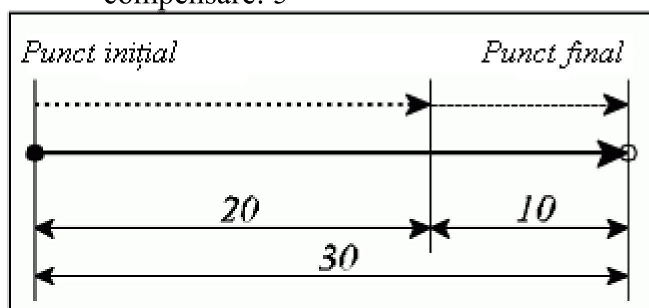


Fig. 14.4-6

Cu programarea G48 (descreştere cu dublul valorii ofsetului)

a. comandă de deplasare: 20
compensare: 5

cazurile b, c, d sunt similare cu cele de la G45

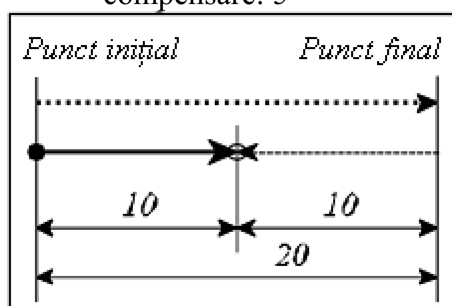


Fig. 14.4-7

În cazul în care după comanda G45…G48 în bloc se află comenzi de deplasare pentru câteva axe, compensarea rezultantă va fi activă pentru fiecare axă programată separat cu valoarea specificată la D (generată non vectorial).

Dacă de exemplu D1=30, comanda G91 G45 G1 X100 Y40 D1 va produce deplasările x=130, y=70.

Compensările rezultante nu pot fi şterse cu o comandă G comună (de exemplu G49 pentru compensarea lungimii) sau prin programarea D00, ci doar cu o comandă G45…G48 cu semnificație opusă.

În folosirea G45…G48, se poate aplica doar un cod D, altfel se va afişa mesajul de eroare *3008 ERRONEOUS G45…G48 (cod G45…G48 eronat)*.

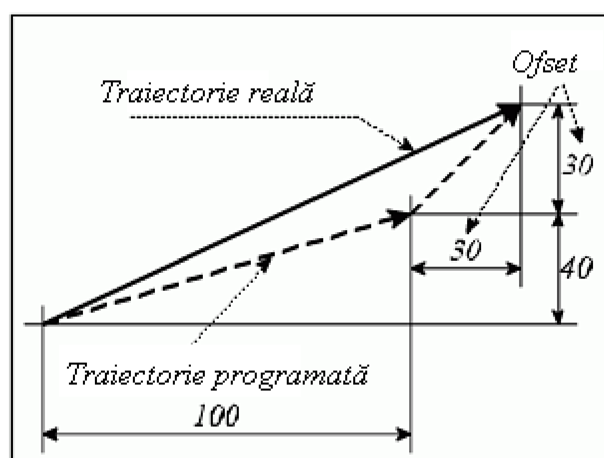


Fig. 14.4-8

Dacă este programată o deplasare incrementală 0 împreună cu o comandă G45…G48, un semn precedând zeroul va fi de asemenea interpretat de către comanda numerică după cum urmează: dacă D1=12

Comandă NC	G45 XI0 D1	G46 XI0 D1	G45 XI-0 D1	G46 XI-0 D1
deplasament	x=12	x=-12	x=-12	x=12

O compensare a razei sculei aplicată cu unul din codurile G45…G48 este de asemenea aplicabilă la ¼ şi ¾ din cercuri, în cazul în care centrele cercurilor sunt specificate la adresele I, J sau K.

Un exemplu: D1=10

```
N1 G91 G46 G0 X40 Y40 D1
N2 G47 G1 Y100 F180
N3 G47 X40
N4 Y-40
N5 G48 X60
N6 Y40
N7 G47 X20
N8 G45 Y-0
N9 G46 G3 X40 Y-40 I40
N10 G45 G1 X0
N11 G45 Y-20
N12 G45 G2 X-40 Y-40 I-40
N13 G45 G1 X-120
N14 G46 G0 X-40 Y-40
```

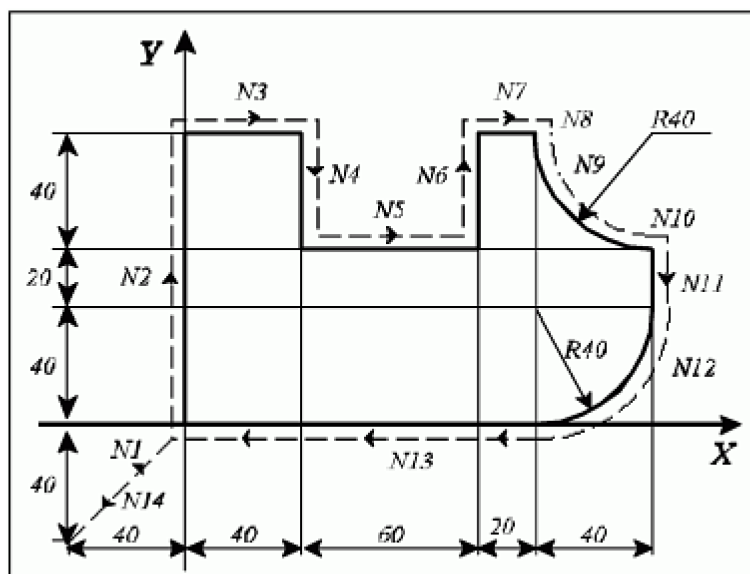


Fig. 14.4-9

14.5 Compensarea sculei (G38, G39, G40, G41, G42)

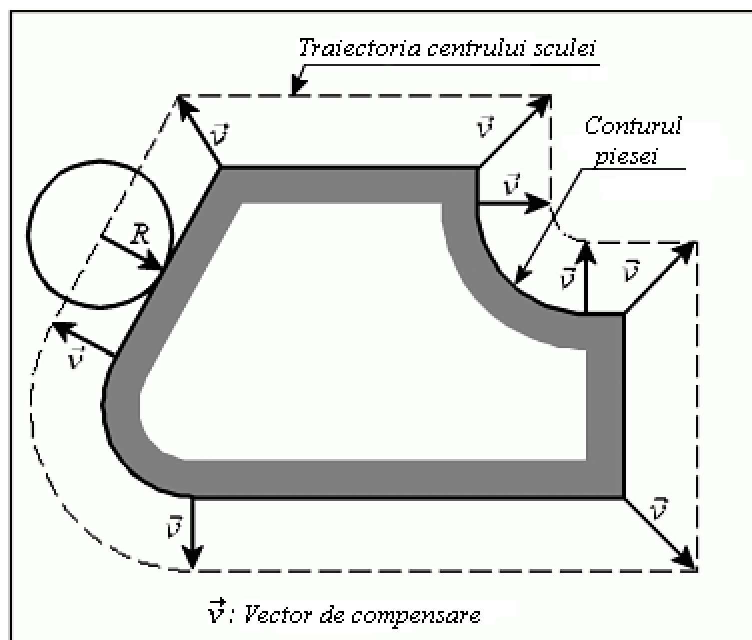


Fig. 14.5-1

Pentru a se putea freza conturul unei piese cu două dimensiuni specificând în program punctele acelui contur la fel ca în desen (indiferent de dimensiunea sculei folosite), comanda numerică trebuie să conducă centrul sculei pe o traiectorie paralelă cu conturul programat şi distanţată cu mărimea razei sculei. Comanda numerică va determina distanţa dintre traiectoria centrului sculei şi conturul programat în concordanţă cu valoarea de compensare a razei sculei la care se face referire prin numărul de compensare D.
Vectorul de compensare este un vector cu două dimensiuni, calculat de către comanda numerică în fiecare bloc, modificându-se deplasările programate cu vectorii de compensare calculaţi la începutul şi sfârşitul fiecărui bloc. Lungimea şi direcţia fiecărui vector de compensare obţinut variază cu valoarea de compensare (apelată la adresa D) şi cu geometria de tranziţie dintre cele două blocuri.
Vectorii de compensare sunt calculaţi în planul selectat cu instrucţiunile G17, G18, G19. Acesta este planul de compensare a sculei. Deplasările în afara acestui plan nu sunt influenţate de compensare. Dacă, de exemplu, este selectat planul X, Y în starea G17, vectorii de compensare vor fi calculaţi în acest plan. În acest caz orice mişcare în direcţia Z nu va fi afectată de compensare.
Planul de compensare nu poate fi schimbat în timpul calculării unei compensări de rază. Orice încercare de a face aceasta va determina afişarea mesajului de eroare *3010 PLANE SELECT. IN G41, G42 (selectare de plan în G41, G42)*.
Dacă trebuie definit un plan de compensare cu axele adiţionale,acestea trebuie definite ca axe paralele în parametri. Dacă, de exemplu, U este presupusă ca axă paralelă şi compensarea razei sculei trebuie aplicată în planul Z, U , planul poate fi selectat prin specificarea G18 U_Z_.

G40: anulare compensare sculă
G41: compensare sculă la stânga
G42: compensare sculă la dreapta

Comanda G41 sau G42 va impune calcularea compensării. În starea G41 sau G42 contururile programate vor fi urmărite din partea stângă sau respectiv din partea dreaptă (văzut din direcţia de mişcare). Numărul de compensare al razei sculei trebuie să fie specificat la adresa D.

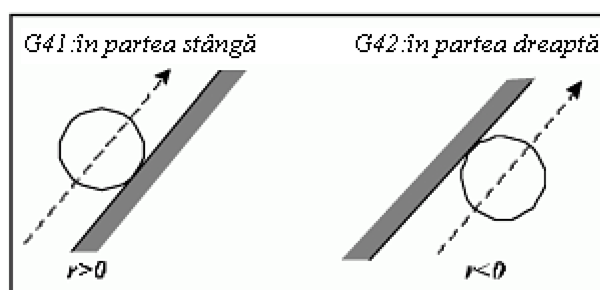


Fig. 14.5-2

Specificarea D00 este întotdeauna echivalentă cu apelarea valorii zero a razei. Calculele compensărilor sunt executate pentru mişcările de interpolare G00, G01, G02, G03.

Punctele de mai sus se referă la specificarea compensării pozitive a razei sculei, dar valoarea ei poate fi şi negativă. Aceasta are o semnificaţie practică dacă, de exemplu, un anumit subprogram trebuie să fie utilizat pentru definirea conturului unei piese cuprinzătoare („mamă“) şi a piesei cuprinse („tată“), care sunt împerecheate. Un mod posibil de a face acest lucru este să se prelucreze piesa cuprinzătoare cu G41 şi piesa cuprinsă cu G42. Totuşi, această conversie poate fi omisă din program atunci când piesa cuprinzătoare este prelucrată cu o compensare de rază pozitivă şi piesa cuprinsă cu o compensare de rază negativă. Acum traiectoria centrului sculei este inversată în raport cu cea programată cu G41 sau G42.

	Compensarea razei: **Pozitivă**	Compensarea razei: **Negativă**
G41	în partea stângă	în partea dreaptă
G42	în partea dreaptă	în partea stângă

Notă:

- pentru simplificare, descrierile şi figurile care urmează se vor referi întotdeauna la compensare de rază pozitivă.

Comanda G40, sau D00 va anula compensarea de ofset. Diferenţa între cele două comenzi este că D00 şterge numai vectorul de compensare, lăsând starea G41 sau G42 neschimbată. Dacă după aceea se face o referire la o adresă D diferită de 0, vectorul de compensare va fi calculat cu noua rază a sculei ca rezultat al funcţiei stării G41 sau G42.
Dacă se foloseşte instrucţiunea G40, orice referire la adresa D nu va avea efect până când nu se programează G41 sau G42.
Procedura de impunere şi anulare a compensării de rază este detaliată în paragrafele care vor urma.
Comenzile G40, G41, G42 sunt modale. Comanda numerică va trece în starea G40 după punerea sub tensiune, la terminarea programului sau în eventualitatea resetării programului, şi în aceste condiţii vectorul de compensare este şters.
Instrucţiunile de compensare a razei vor fi executate de către comanda numerică doar în modul automat. Ele nu sunt active la programarea unui bloc singular în modul manual. Motivul pentru aceasta este după cum urmează. Pentru a putea calcula vectorul de compensare în punctul final al blocului (interpolare), comanda numerică trebuie să citească şi blocul următor care conţine mişcarea în planul selectat. Vectorul de compensare depinde de tranziţia între cele două interpolări. În consecinţă, trebuie să fie pre-procesate câteva blocuri (interpolări) pentru calculul unui vector de compensare.

Înainte de a începe discuția asupra detaliilor calculului compensării, trebuie introdusă o dată auxiliară. Este vorba de „α“, unghiul de colț între două blocuri consecutive, văzut dinspre partea piesei. Direcția lui α este funcție de sensul în care merge scula împrejurul colțului: de la stânga sau de la dreapta.

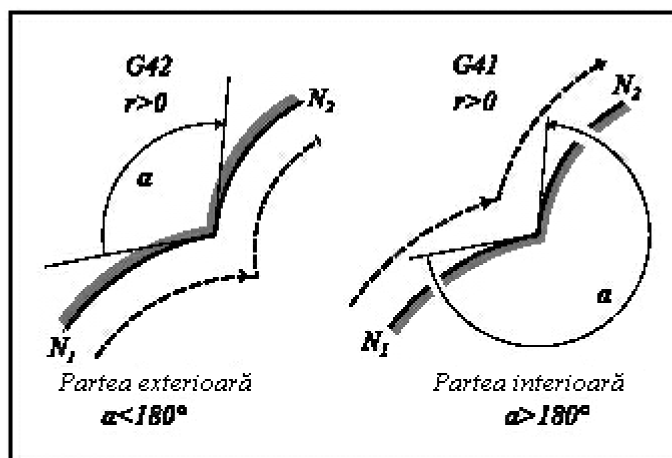


Fig. 14.4-5

Comanda numerică va selecta strategia de a merge în jurul punctelor de intersecție în funcție de unghiul α. Dacă α >180° , adică scula lucrează în interior, comanda numerică va calcula un punct de intersecție între cele două interpolări. Dacă α <180° , adică scula se mişcă prin exterior, comanda numerică *poate adăuga* în continuare secțiuni drepte.

14.5.1 Pornirea compensării sculei

După punerea sub tensiune, sfârşitul unui program sau resetarea la început a unui program, comanda numerică va activa starea G40. Vectorul de ofset va fi şters, iar traiectoria centrului sculei va coincide cu traiectoria programată.
Ca efect al instrucțiunilor G41 sau G42 comanda numerică va intra în modul de calcul al compensării razei. Valoarea compensării va fi preluată din celula de compensare (registrul D). Se va trece în starea G41 sau G42 doar într-un bloc care conține o interpolare liniară (G00 sau G01). Comanda numerică va afişa mesajul de eroare *3043 G41, G42 IN G2, G3* la orice încercare de a impune calculul compensării într-o interpolare circulară (G02, G03).
Comanda numerică va alege să pornească procedura de calcul a compensării sculei dacă a fost comandat G41 sau G42 după G40. Cu alte cuvinte, comanda numerică nu va porni procedura atunci când compensarea este ştearsă cu D00 şi reactivată cu Dnn (nn fiind un număr diferit de 0).
Situațiile de bază când se porneşte compensarea depind de unghiul α de la colțul a două blocuri consecutive şi de tipul interpolărilor (liniar la liniar, liniar la circular), cum se arată mai jos. Figurile se referă la cazul G42, presupunând compensarea pozitivă a razei.

Notă: simbolurile din figurile următoare au următoarele semnificații :

r: valoarea compensării razei,
L: linie dreaptă,
C: arc de cerc,
S: punct de oprire bloc singular,
Linie punctată: traiectoria centrului sculei,
Linie continuă: traiectoria programată.

<u>Situațiile de bază când se porneşte compensarea sculei</u>

```
(G40)                (G40)
G42 G01 X_Y_D_       G40 G01 X_Y_D_
X_Y_                 G2 X_Y_R_
```

Mergând prin interiorul unui colț 180° <α<360°

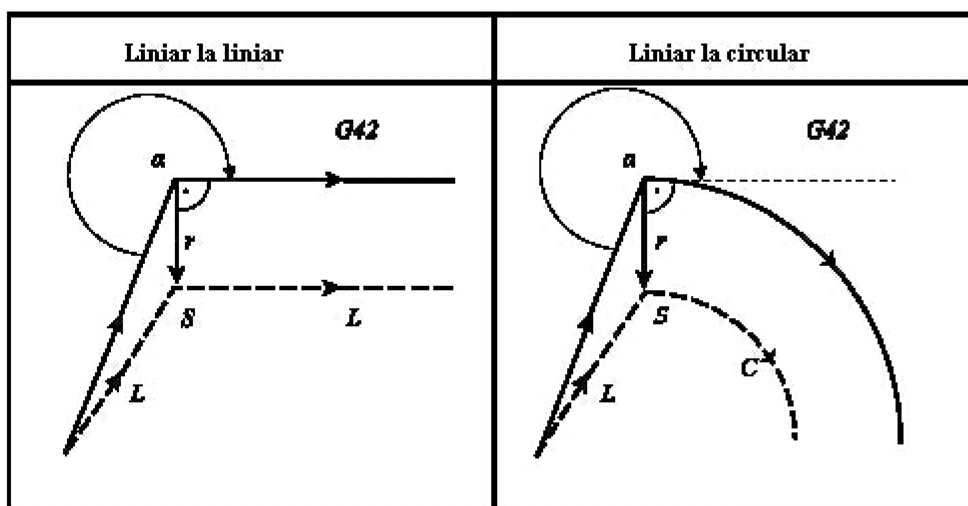


Fig. 14.5.1-1

Mergând prin exteriorul unui colț cu unghi obtuz 90° ≤α ≤180°

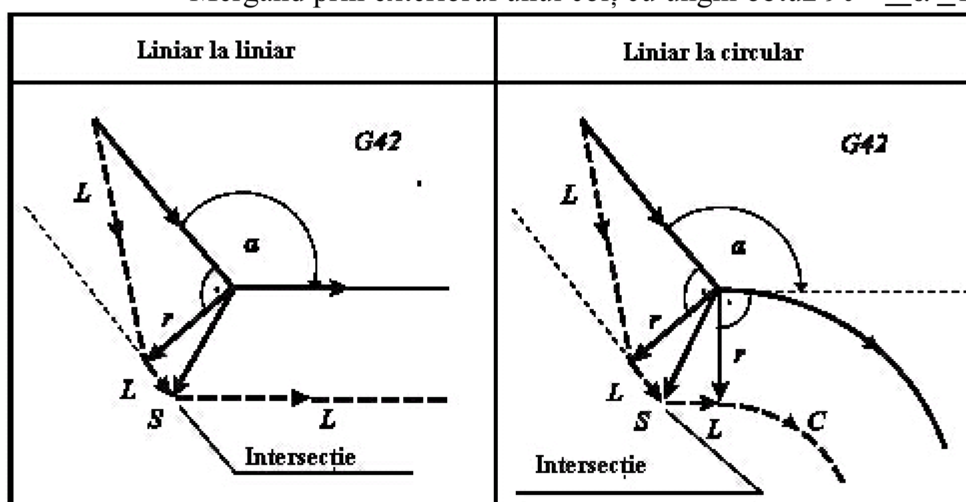


Fig. 14.5.1-2

Mergând prin exteriorul unui colț cu unghi ascuțit 0° ≤α <90°

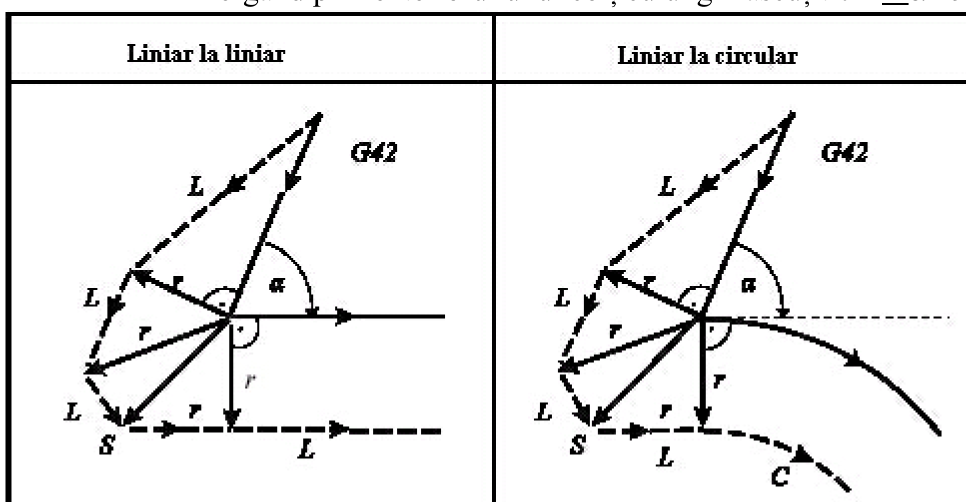


Fig 15.5.1-3

Situațiile speciale când se pornește compensarea razei:

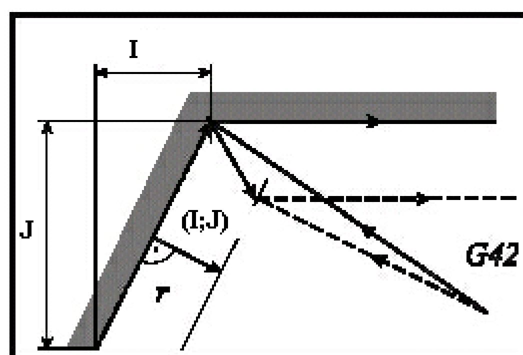


Fig. 14.5.1-4

Dacă sunt stabilite valori pentru I, J, K în blocul de selectare a compensării (G41, G42) – dar numai pentru cele din planul selectat (de exemplu, pentru I, J în cazul lui G17) – comanda numerică se va deplasa pentru pornirea compensării razei în punctul de intersecție dintre blocul următor şi linia dreaptă definită de I, J, K. Valorile pentru I, J, K sunt întotdeauna incrementale, vectorul definit prin ele fiind îndreptat către punctul de sfârşit al interpolării, în care a fost programat. Această facilitate este folositoare de exemplu la deplasarea într-un colț interior.

```
...
G91 G17 G40
...
N110 G42 G1 X-80 Y60 I50 J70 D1
N120 X100
...
```

În acest caz comanda numerică va calcula întotdeauna un punct de intersecție indiferent dacă este prelucrat un colț interior sau unul exterior.

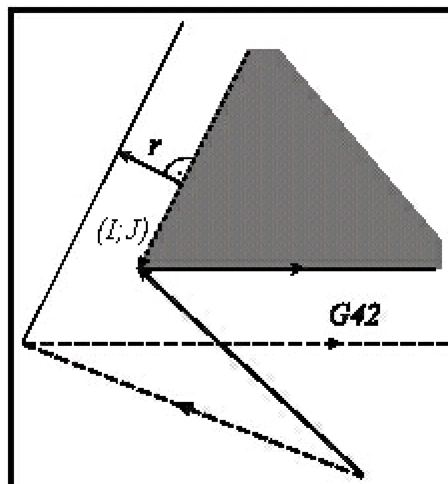


Fig. 14.5.1-5

Dacă nu este găsit un punct de intersecție, comanda numerică se va deplasa, în unghiuri drepte, în punctul de pornire al următoarei interpolări.

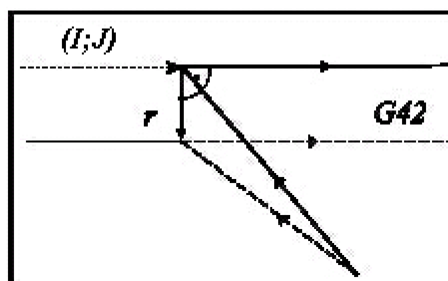


Fig. 14.5.1-6

Atunci când se impune compensarea printr-un bloc special în care nu este programată nici o mişcare în planul selectat, compensarea va fi impusă fără nici o deplasare, lungimea calculată a vectorului de compensare este 0. Vectorul de compensare este calculat la sfârşitul următorului bloc de mişcare, funcție de strategia corespunzătoare compensării în modul ofset (vezi capitolul următor).

```
...
N10 G40 G17 G0 X0 Y0
N15 G42 D1
N20 G1 X80
N25 X110 Y60
...
```

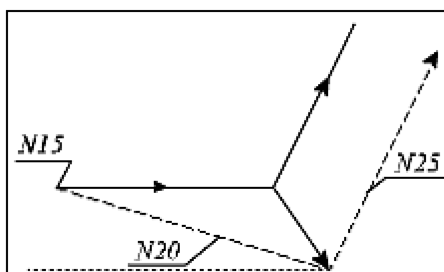


Fig. 14.5.1-7

Dacă este programat deplasament 0 (sau acesta rezultă) în blocul care conține activarea compensării (G41, G42), comanda numerică nu va efectua nici o mişcare ci va continua prelucrarea conform cu strategia mai sus menționată.

```
...
N10 G40 G17 G0 X0 Y0
N15 G91 G42 D1 X0
N20 G1 X80
N25 X30 Y60
...
```

Dacă se obține un deplasament 0 în planul selectat în blocul care urmează pornirii compensării, vectorul de compensare va fi impus în unghi drept la interpolarea care a executat impunerea. Traiectoria sculei în următoarea interpolare nu va fi paralelă cu conturul programat:

```
...
N10 G40 G17 G0 X0 Y0
N15 G91 G42 D1 X80
N20 G1 X0
N25 X30 Y60
N30 Y60
...
```

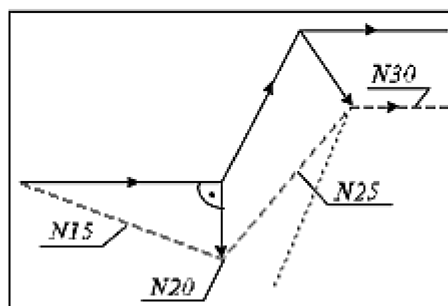


Fig. 14.5.1-8

14.5.2 Reguli de compensare a sculei în modul ofset

În modul ofset, vectorii de compensare vor fi calculați în mod continuu între blocurile de interpolare G00, G01, G02, G03 (vezi situațiile de bază) până cândva fi inserat cel puțin un bloc care nu conține deplasamente în planul selectat. Această categorie include un bloc conținând întârziere sau funcții.

Situațiile de bază ale modului ofset

Calculul punctului de intersecție pentru colț interior 180° <α<360°

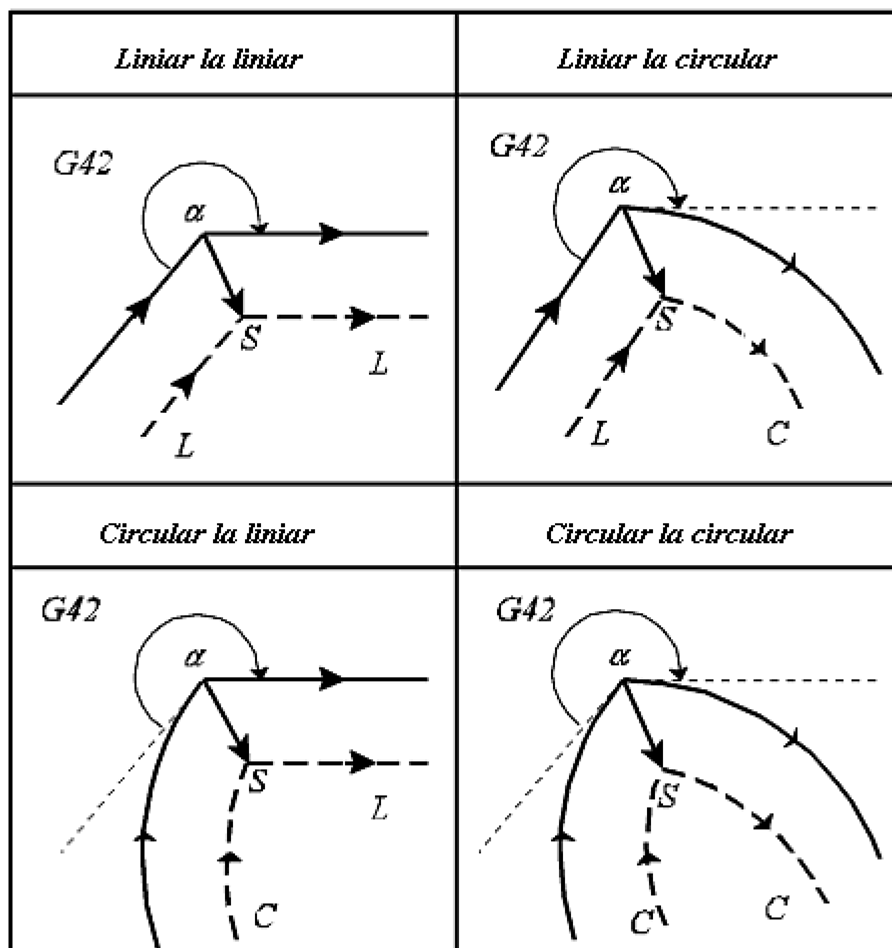


Fig. 14.5.2-1

Se poate întâmpla să nu se obțină un punct de intersecție cu anumite valori ale razei vârfului sculei. În acest caz comanda numerică se opreşte din execuția interpolării precedente şi afişează mesajul de eroare *3046 NO INTERSECTION G41, G42 (G41, G42 nu se intersectează).*

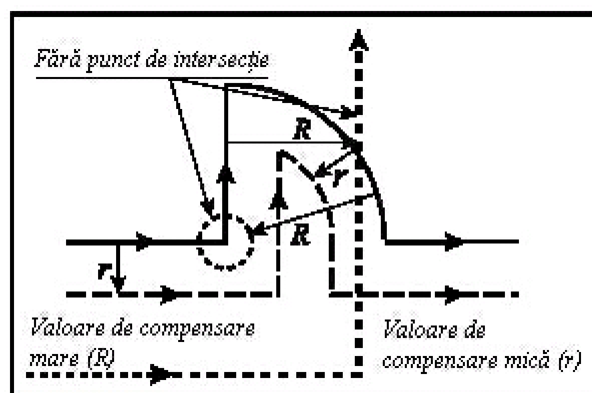


Fig. 14.5.2-2

Mergând prin exteriorul unui colț cu unghi obtuz $90° \leq \alpha \leq 180°$

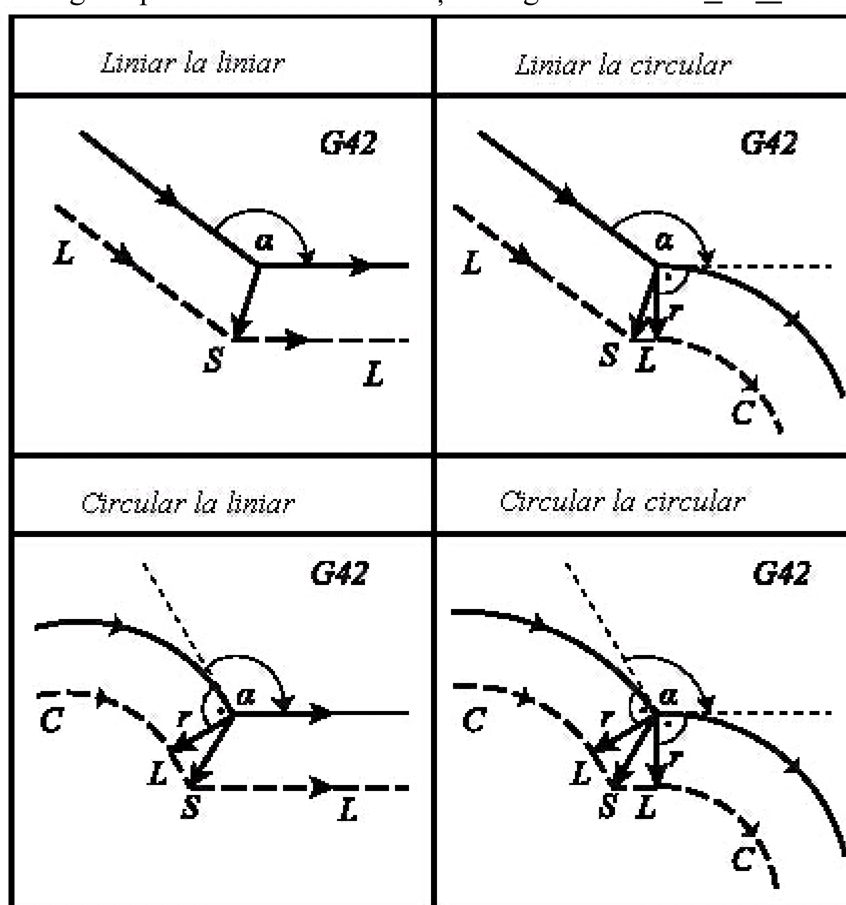


Fig. 14.5.2-3

Mergând prin exteriorul unui colț cu unghi ascuțit 0° $\leq \alpha < 90°$

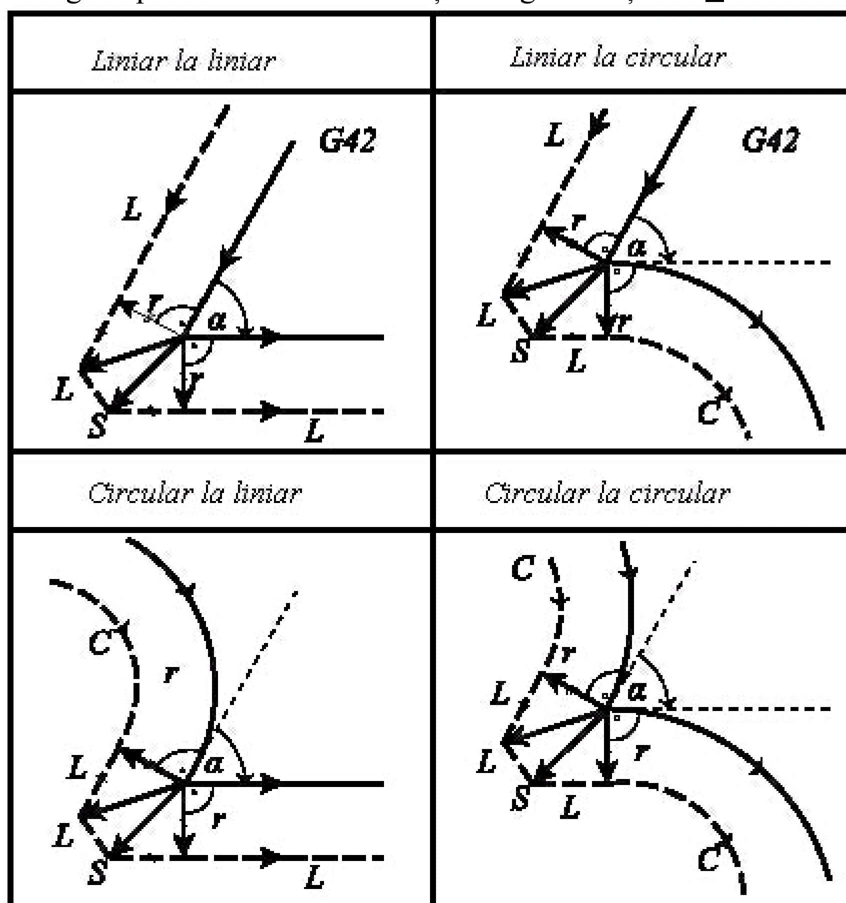


Fig. 14.5.2-4

Situațiile speciale ale modului ofset

Dacă este programat deplasament 0 (sau așa este obținut) în planul selectat într-un bloc în modul ofset, va fi poziționat un vector perpendicular în punctul final al interpolării precedente, lungimea vectorului va fi egală cu compensarea razei. Situațiile de acest fel trebuie tratate cu precauție datorită riscului de a tăia accidental mai puțin sau distorsionat (în cazul unui cerc).

De exemplu:

```
...G91 G17 G42. . .
N110 G1 X40 Y50
N120 X0
N130 Z90
N140 X50 Y-20
...
```

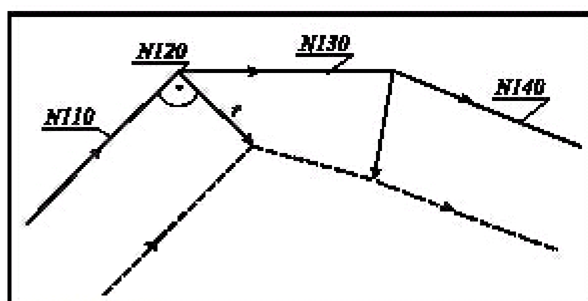


Fig. 14.5.2-5

14.5.3 Anularea modului ofset

Comanda G40 va anula calculul compensării razei sculei. O astfel de comandă se poate da numai pentru interpolarea liniară. Dacă se încearcă programarea G40 într-o interpolare circulară comanda numerică va afişa mesajul de eroare *3042 G40 IN G2, G3.*

Situații de bază de anulare a modului ofset

```
(G42)                (G42)
G01 X_Y_             G02 X_Y_R_
G40 X_Y_             G40 G1 X_Y_
```

Mergând prin interiorul unui colț 180° < α < 360°

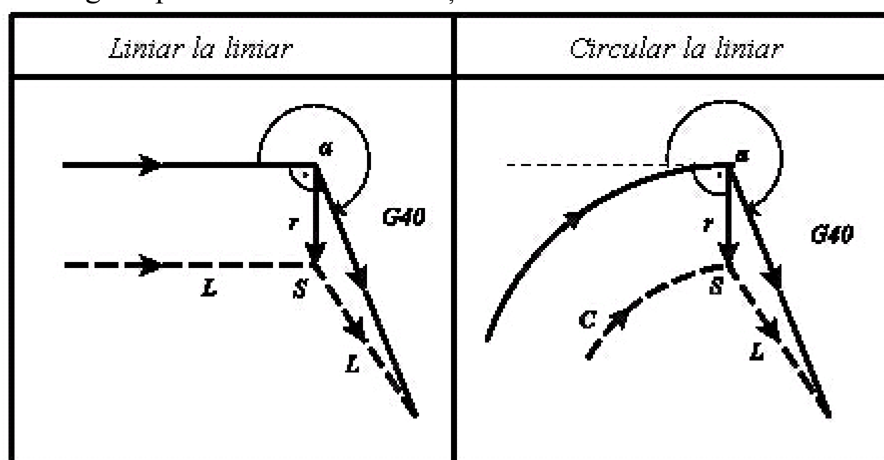


Fig. 14.5.3-1

Mergând prin exteriorul unui colț cu unghi obtuz 90° ≤ α ≤ 180°

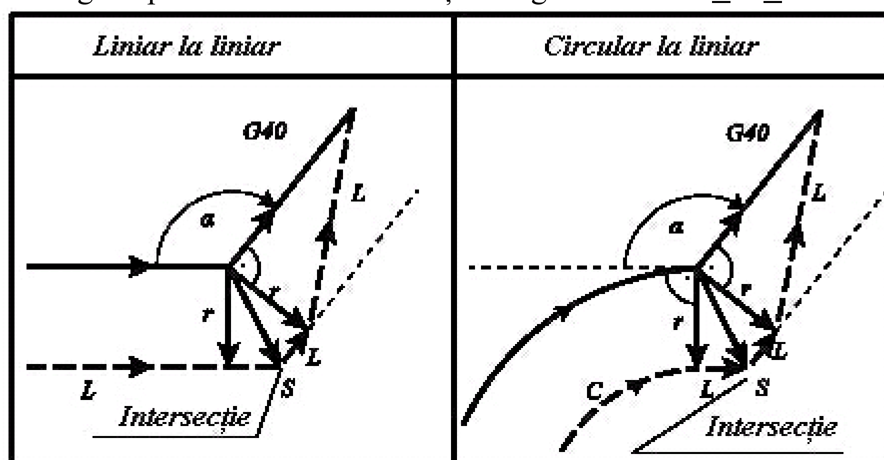


Fig. 14.5.3-2

Mergând prin exteriorul unui colț cu unghi ascuțit 0° ≤ α < 90°

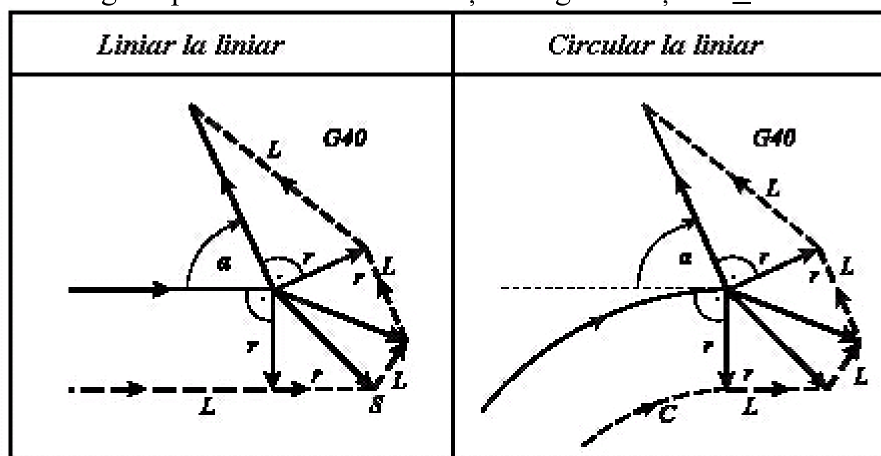


Fig. 14.5.3-3

Situațiile speciale de anulare a modului ofset

Dacă sunt atribuite valori pentru I, J, K în blocul de anulare a compensării (G40) – dar numai acelora din planul selectat (de exemplu pentru I, J în cazul lui G17) – comanda numerică va deplasa scula la punctul de intersecție dintre interpolarea precedentă şi linia dreaptă definită de I, J, K. Valorile pentru I, J, K sunt întotdeauna incrementale, vectorul definit prin acestea pornind din punctul final al interpolării precedente.
Această facilitate este utilă de exemplu pentru deplasarea dintr-un colț interior.

```
...
...G91 G17 G42...
N100 G1 X50 Y60
N110 G40 X70 Y-60 I100 J-20
...
```

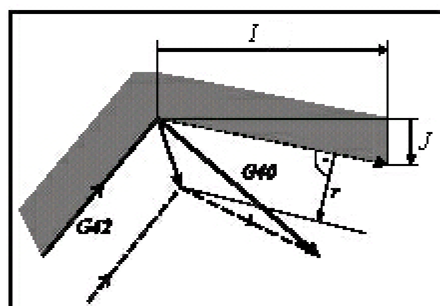


Fig. 14. 5.3-4

În acest caz, comanda numerică va calcula întotdeauna un punct de intersecție indiferent dacă trebuie prelucrat un colț interior sau exterior.

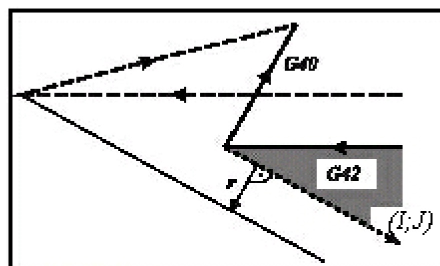


Fig. 14.5.3-5

Dacă nu este găsit un punct de intersecţie, comanda numerică se va deplasa, în unghi drept, în punctul final al precedentei interpolări.

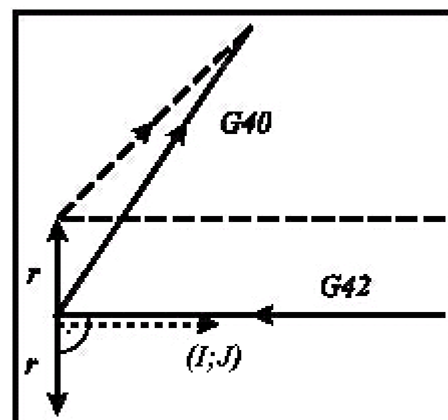


Fig. 14.5.3-6

Dacă compensarea este anulată într-un bloc în care nu este programată nici o deplasare în planul selectat, va fi impus un vector de ofset perpendicular în punctul final al precedentei interpolări şi vectorul de compensare va fi şters la sfârşitul următorului bloc de mişcare.

```
...G42 G17 G91...
N110 G1 X80 Y40
N120 G40
N130 X-70 Y20
...
```

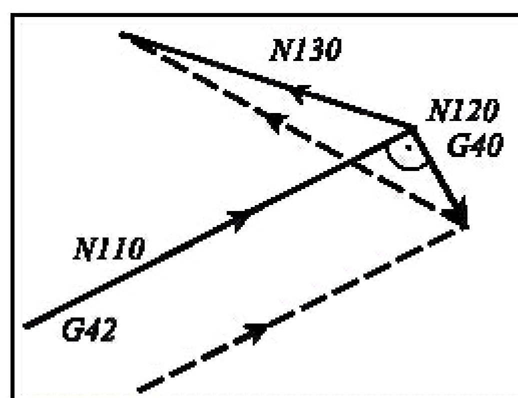


Fig. 14.5.3-7

Dacă este programat deplasament 0 (sau acesta rezultă) în blocul care conţine anularea compensării (G40), va fi calculat un vector de ofset perpendicular în punctul final al precedentei interpolări, şi comanda numerică va termina procedura în blocul G40.
De exemplu:

```
...G42 G17 G91...
N110 G1 X80 Y40
N120 G40 X0
N130 X-70 Y20
...
```

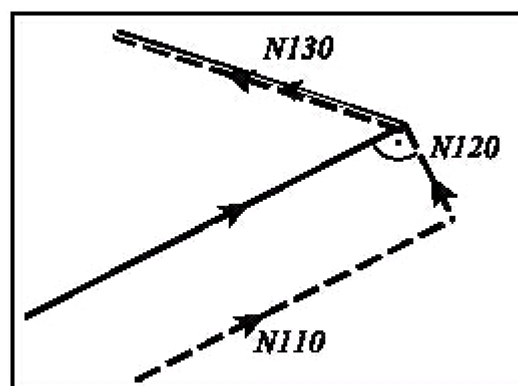


Fig. 14.5.3-8

14.5.4 Schimbarea direcției ofsetului în modul ofset

Direcția de calcul a compensării razei sculei este dată în tabelul următor:

	Compensarea razei: **pozitivă**	Compensarea razei: **negativă**
G41	stânga	dreapta
G42	dreapta	stânga

Direcția ofsetului poate fi inversată chiar în timpul calculului compensării razei sculei. Aceasta se poate realiza prin programarea G41 sau G42, sau prin apelarea unei compensări a razei sculei de semn opus la adresa D. Când se inversează direcția ofsetului, comanda numerică nu va face verificarea dacă aceasta este „în afară“, va calcula întotdeauna un punct de intersecție în primii paşi. Figurile următoare presupun valori pozitive pentru raza sculei şi treceri de la G41 la G42.

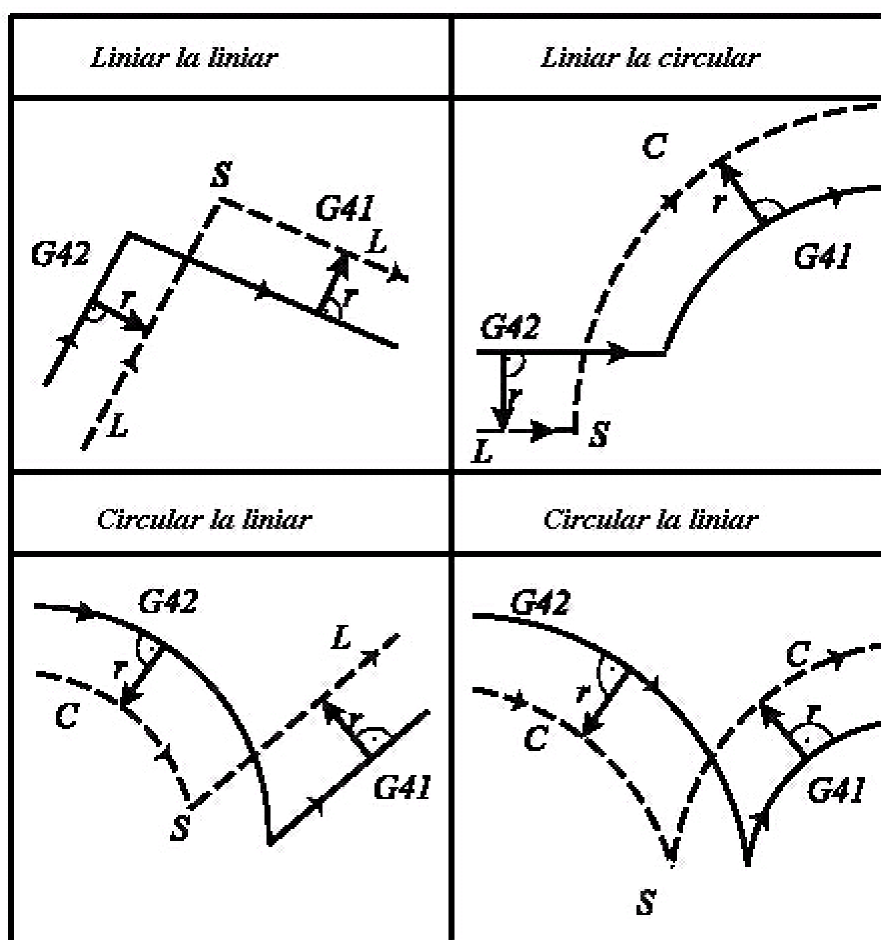


Fig. 14.5.4-1

Dacă nu este găsit un punct de intersecție pentru o tranziție liniar-la-liniar, traiectoria sculei va fi conform figurii alăturate:

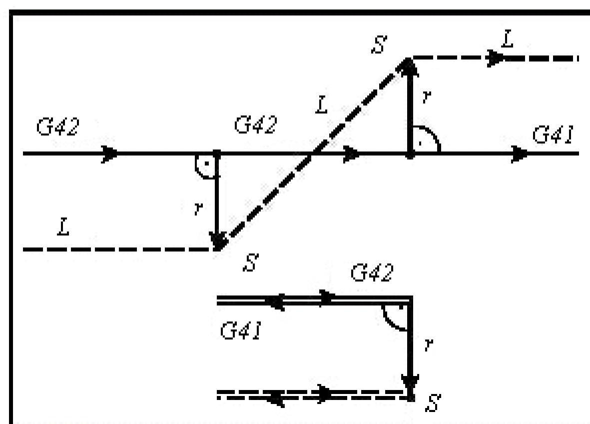


Fig. 14.5.4-2

Dacă nu este găsit un punct de intersecție pentru o tranziție liniar-la-circular, traiectoria sculei va fi conform figurii alăturate:

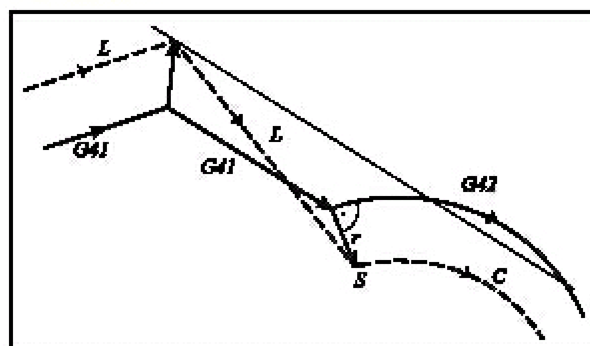


Fig. 14.5.4-3

Dacă nu este găsit un punct de intersecție pentru o tranziție circular-la-liniar sau circular-la-circular, sfârşitul vectorului de compensare în punctul de pornire al primei interpolări circulare va fi conectat cu punctul final al vectorului de compensare perpendicular în punctul de pornire al celei de a doua interpolări cu un arc circular cu raza programată R necorectată. Centrul arcului de cerc de interconectare nu va coincide cu centrul arcului programat. Comanda numerică va afişa mesajul *3047 CHANGE NOT POSSIBLE (schimbarea nu este posibilă)* dacă schimbarea de direcție nu se poate face chiar cu relocarea centrului cercului rezumată mai sus.

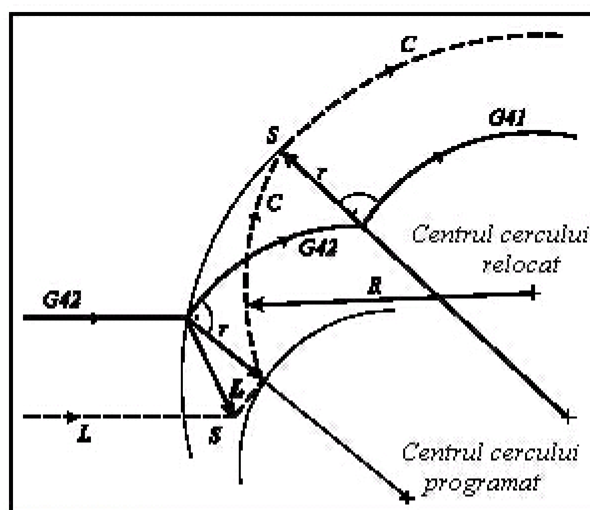


Fig.14.5.4-4

14.5.5 Programarea reținerii vectorului (G38)

Sub acțiunea comenzii

G38 v

comanda numerică va reține vectorul ultimei compensări între interpolarea precedentă şi blocul G38 în modul ofset şi îl va executa la sfârşitul blocului G38 independent de tranziția între blocul G38 şi blocul următor. Codul G38 are acțiune singulară, adică nu este modal. G38 trebuie să fie programat în mod repetat dacă vectorul trebuie reținut în câteva blocuri consecutive.
G38 poate fi programat numai în stările G00 sau G01, adică blocul de reținere a vectorului trebuie să fie în mod invariabil o interpolare liniară, altfel comanda numerică va afişa mesajul de eroare *3040 G38 NOT IN G0, G1 (G38 nu este în bloc G0, G1)*. Dacă G38 nu este folosit în modul ofset (G41, G42), comanda numerică va afişa mesajul de eroare *3039 G38 CODE IN G40 (G38 în bloc G40)*.
Un exemplu de utilizare a G38:

```
...G17 G41 G91...
N110 G1 X60 Y60
N120 G38 X90 Y-40
N130 X20 Y70
N140 X60
...
```

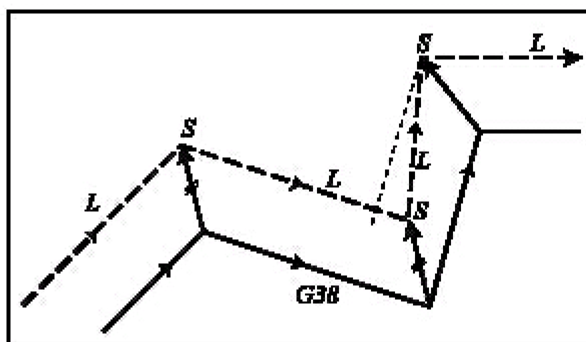


Fig. 14.5.5-1

Pentru a programa o retragere fără să se anuleze modul ofset:

```
...G17 G42 G91...
N110 G1 X40
N120 G38 Z50
N130 G38 Y70
N140 G38 Y-70
N150 X60
...
```

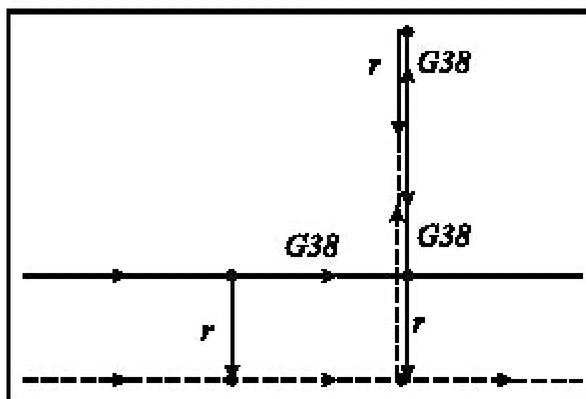


Fig. 14.5.5-2

14.5.6 Programarea arcelor de colț (G39)

Prin programarea

G39 (I J K)

va fi posibil – în modul ofset – să se evite calcularea automată a punctului de intersecție sau a inserării de secțiuni liniare atunci când se merge în jurul colțurilor exterioare şi în loc de aceasta centrul sculei se va deplasa pe un arc circular egal cu raza sculei.
Se va insera un arc egal cu raza sculei în direcția G02 sau G03 în starea G41 sau respectiv G42.

Punctele de început şi de sfârşit ale arcului vor fi date de un vector de lungimea razei sculei, perpendicular în punctul final al traiectoriei interpolării precedente şi respectiv de un vector de lungimea razei sculei, perpendicular în punctul inițial al traiectoriei interpolării următoare. G39 trebuie programat într-un bloc separat.

```
...G17 G91 G41...
N110 G1 X100
N120 G39
N130 G3 X80 Y-80 I80
...
```

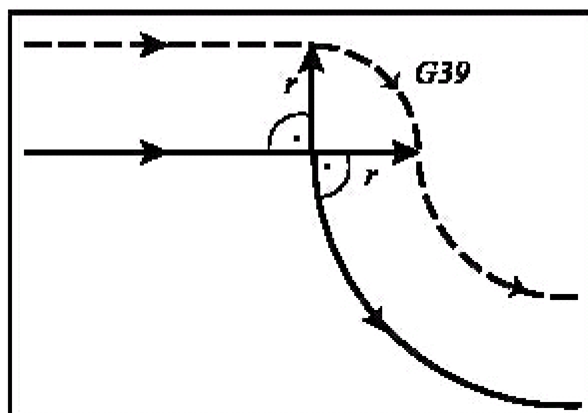


Fig. 14.5.6-1

Când I, J sau K este programat în blocul G39, punctul final al arcului de cerc va fi dat de un vector de lungimea razei sculei, perpendicular pe vectorul definit de I, J sau K, din punctul final al interpolării precedente, în concordanță cu planul selectat.

```
...G17 G91 G41...
N110 G1 X100
N120 G39 I50 J-60
N130 G40 X110 Y30
...
```

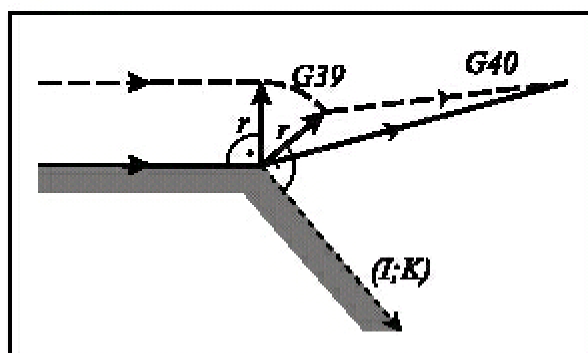


Fig. 14.5.6-2

Pentru comenzile de imagine în oglindă sau rotire selectate anterior este în vigoare vectorul definit de I, J sau K. De fapt, comanda de modificare la scară nu va afecta direcția. În blocul G39 nu poate fi programată nici o comandă de mişcare. Comanda numerică va afişa mesajul de eroare *3036 G39 CODE IN G40 (cod G39 în bloc G40)* dacă se da o comandă G39 în starea G40 sau în modul de compensare 3D.

14.5.7 Informații generale privind aplicarea compensării sculei

În modul ofset (G41, G42), comanda numerică va trebui să calculeze întotdeauna vectorii de compensare dintre două blocuri de interpolare în planul selectat. În practică poate fi necesar să se programeze între două blocuri de interpolare în planul selectat un bloc fără interpolare sau un bloc de interpolare în afara planului selectat. Acestea pot fi:
- funcții (M, S, T)
- întârziere (G4 P)
- interpolare în afara planului selectat ([G18] G1 Z)
- apelare subprogram (M98 P)
- activare sau dezactivare de transformări speciale (G50, G51, G50.1, G51.1, G68, G69)

Notă : La apelarea unui subprogram sunt necesare anumite precauții. În cazul în care subprogramul nu începe cu o mişcare în planul asignat, interpolarea va fi distorsionată.

Comanda numerică va accepta programarea unui **singur** bloc de tipul de mai sus între două blocuri de interpolare din program, păstrând traiectoria sculei neafectată:

```
...G17 G42 G91...
N110 G1 X50 Y70
N120 G4 P2
N130 X60
...
```

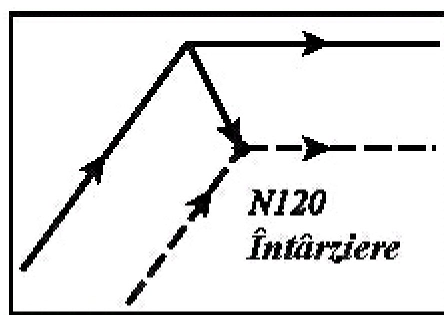


Fig. 14.5.7-1

Atunci când comanda numerică inserează una sau mai multe linii drepte între două interpolări când se deplasează în jurul unui colț, orice alt bloc fără deplasare sau cu deplasare în afara planului selectat, programat între interpolări va fi executat la punctul final al primei interpolări (indicat cu „S“ în figuri).

Atunci când sunt scrise în program două interpolări în afara planului selectat sau două blocuri care nu conțin interpolări, comanda numerică va impune un vector de ofset perpendicular în punctul final al ultimei interpolări executate în planul selectat şi traiectoria va fi distorsionată:

```
...G17 G42 G91...
N110 G1 X50 Y70
N120 G4 P2
N130 S400
N140 X60
...
```

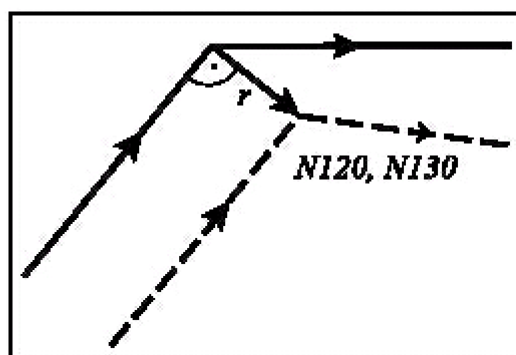


Fig. 14.5.7-2

Dacă nu se poate face o prelucrare pe direcția **Z** fără ca să fie impusă compensarea razei, se poate adopta următoarea procedură:

```
...G17 G91...
N110 G41 G0 X50 Y70 D1
N120 G1 Z-40
N130 Y40
...
```

Acum scula va avea o traiectorie corectă după cum se arată în figura alăturată.

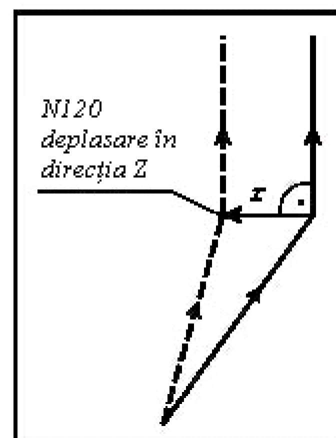


Fig 14.5.7-3

Dacă totuşi mişcarea în direcția **Z** este fragmentată în două părți (avans rapid şi avans de lucru), traiectoria va fi distorsionată deoarece sunt două interpolări consecutive în afara planului selectat:

```
...G17 G91...
N110 G41 G0 X50 Y70 D1
N120 Z-35
N130 G1 Z-5
N130 Y40
...
```

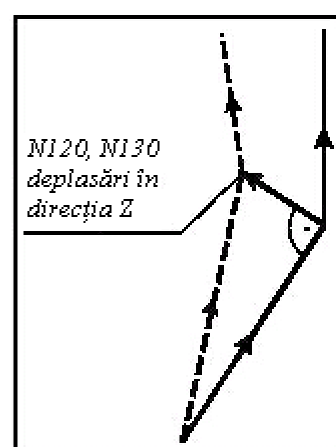


Fig. 14.5.7-4

Pentru a evita aceasta, se introduce o mică deplasare în direcția **Y** între cele două deplasări în direcția **Z**:

```
...G17 G91...
N110 G41 G0 X50 Y69 D1
N120 Z-35
N130 Y1
N140 G1 Z-5
N150 Y40
...
```

Prin aceasta se va obține un vector de compensare corect.

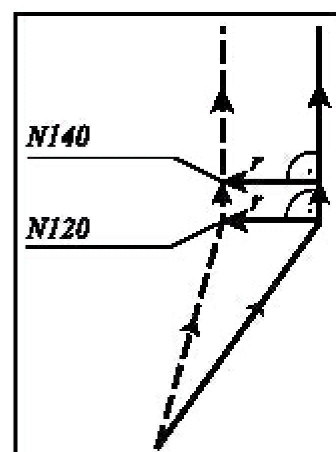


Fig. 14.5.7-5

În continuare se va descrie traiectoria sculei atunci când între două interpolări se inserează instrucțiunile:

G22, G23, G52, G54-G59, G92
G53
G28,G29,G30

Atunci când se programează comanda G22, G23, G52, G54-G59 sau G92 în modul ofset între două blocuri de interpolare, vectorul de compensare va fi şters la punctul final al interpolării anterioare, comanda va fi executată şi vectorul va fi reconstituit la punctul final al următoarei interpolări. Dacă interpolarea anterioară sau cea care urmează este o interpolare circulară, va fi afişat mesajul de eroare *3041 AFTER G2, G3 ILLEG. BLOCK (bloc nepermis după G2, G3)*.

De exemplu:

```
...G91 G78 G41...
N110 G1 X80 Y-50
N120 G92 X0 Y0
N130 X80 Y50
...
```

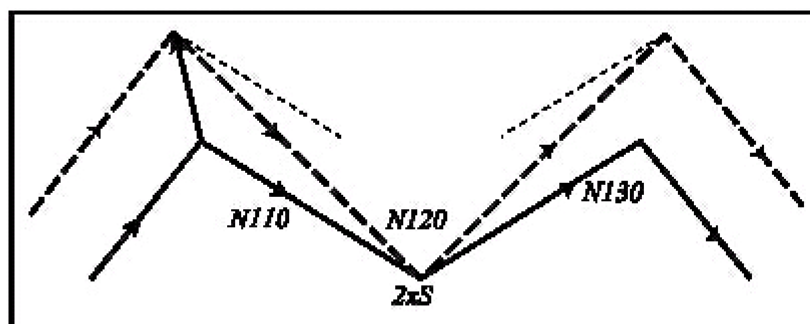


Fig. 14.5.7-6

Dacă este programată comanda G53 în modul ofset între două interpolări, vectorul de compensare va fi şters la punctul final al blocului anterior, se va executa poziționarea în G53 şi vectorul va fi reconstituit la punctul final al următoarei interpolări (alta decât G53). Dacă interpolarea anterioară sau cea care urmează este o interpolare circulară, va fi afişat mesajul de eroare *3041 AFTER G2, G3 ILLEG. BLOCK (bloc nepermis după G2, G3)*.

De exemplu:

```
...G91 G17 G41...
N110 G1 X80 Y-50
N120 G53 Y80
N130 G53 Y0
N140 X80 Y50
...
```

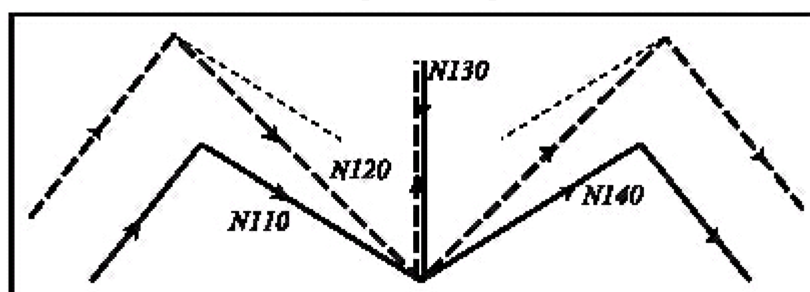


Fig. 14.5.7-7

Dacă este programată comanda G28 sau G30 (urmată de G29) între două blocuri în modul ofset, vectorul de compensare va fi şters la punctul final al blocului care poziționează scula în punctul intermediar, scula se va mişca în punctul de referință şi vectorul va fi reconstituit la punctul final al blocului de revenire G29.

De exemplu:

```
...G91 G17 G41...
N110 G1 X80 Y-50
N120 G28 Y80
N130 G29 Y0
N140 X80 Y50
...
```

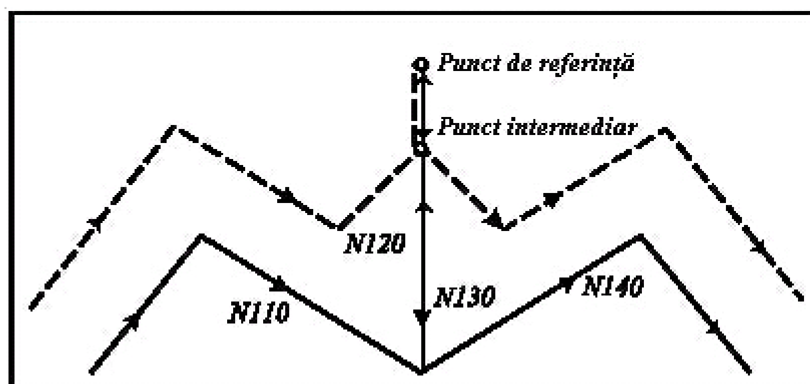


Fig. 14.5.7-8

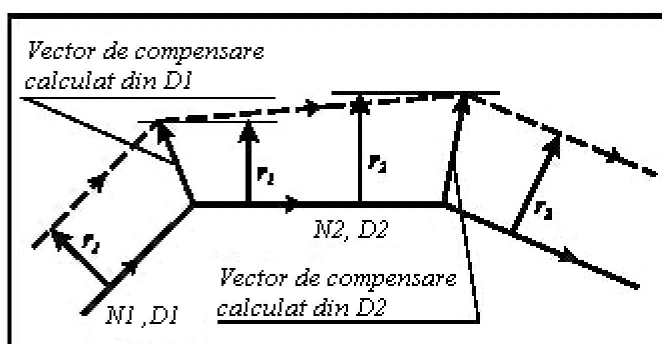


Fig. 14.5.7-9

În modul ofset se poate de asemenea apela o valoare nouă de compensare la adresa D. În eventualitatea inversării semnului razei, direcția de mişcare de-a lungul conturului va fi inversată (vezi şi mai înainte). Altfel, va fi aplicabilă următoarea procedură. Vectorul de compensare va fi calculat cu noua valoare a razei la punctul final al interpolării, în care a fost programată noua adresă D. Deoarece vectorul de compensare a fost calculat cu valoarea anterioară a razei în punctul de început al acelui bloc, traiectoria centrului sculei nu va fi paralelă cu traiectoria programată.
De asemenea, o valoare nouă pentru compensarea razei poate fi apelată la adresa D şi într-o interpolare circulară, totuşi, de această dată, centrul sculei se va deplasa de-a lungul unui arc cu rază variabilă.
Un caz special pentru cele discutate anterior este anularea sau impunerea compensării cu D00 sau respectiv Dnn, atunci când suntem în modul ofset. Aveți în vedere diferența între traiectoriile sculei cu referire la următorul exemplu, în care compensarea este impusă cu G41 sau G42 şi anulată cu G40, sau când compensarea este impusă şi anulată prin programarea D.

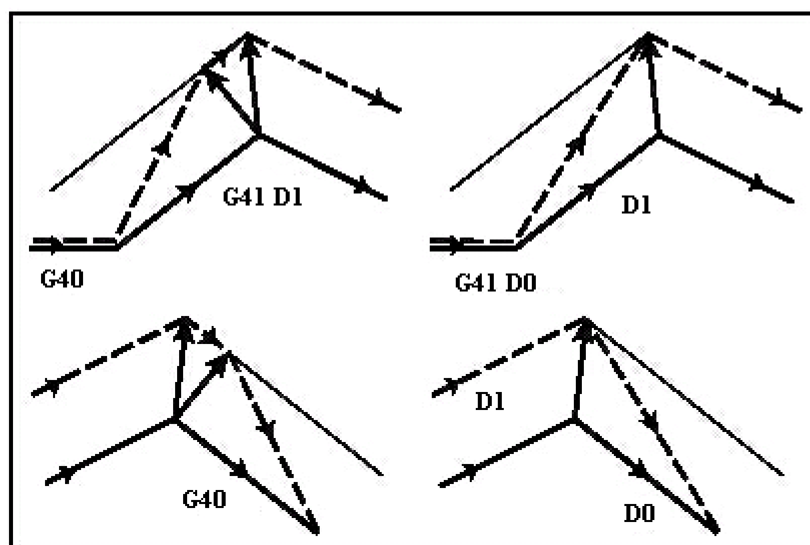


Fig. 14.5.7-10

Se poate folosi de asemenea şi un detaliu de program sau subprogram pentru a prelucra una din piesele unui ansamblu: cea cuprinsă folosind compensare de rază pozitivă, respectiv piesa cuprinzătoare folosind compensare de rază negativă, şi viceversa. Să revedem următorul detaliu de program:

```
...
N020 G42 G1 X80 D1
N030 G1 Z-5
N040 G3 I-80
N050 G1 Z2
N060 G40 G0 X0
...
```

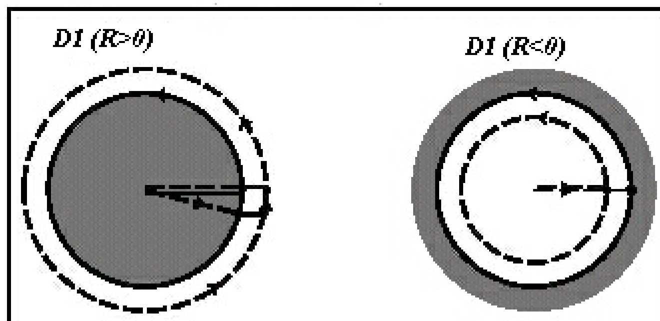
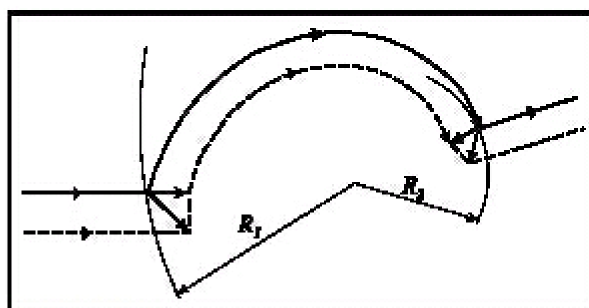


Fig. 14.5.7-11

Când compensarea de rază este aplicată unui cerc cu rază variabilă, comanda numerică va calcula vectorul (vectorii) de compensare pentru un cerc imaginar începând din punctul de start, a cărui rază este egală în punctul de start cu raza cercului programat, iar centrul coincizând cu cel programat. Vectorul (vectorii) de compensare va fi calculat pe un cerc imaginar la punctul final al său, a cărui rază este egală cu raza în punctul final a cercului programat, punctul de centru coincizând cu cel al cercului programat.


Fig 14.5.7-12

Când a fost programat un cerc complet, se poate întâmpla adesea ca traiectoria sculei să acopere mai mult decât un cerc complet în modul ofset.
De exemplu, aceasta se poate întâmpla la programarea unei inversări de direcţie pe un contur:

```
...G17 G42 G91...
N110 G1 X30 Y-40
N120 G41 G2 J-40
N130 G42 G1 X30 Y40
...
```

centrul sculei acoperă un arc de cerc întreg din punctul P_1 în punctul P_1 şi un altul din punctul P_1 în punctul P_2.

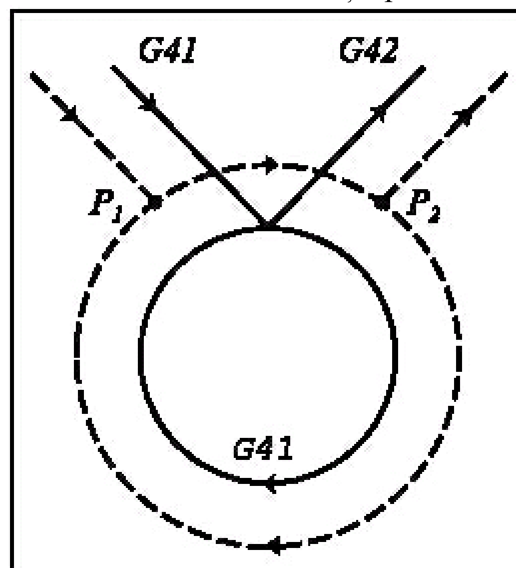


Fig. 14.5.7-13

Când modul ofset este anulat prin programarea I, J, K, va rezulta o condiție similară:

```
...G18 G90 G41...
N090 G1 X30
N100 G2 J-60
N110 G40 G1 X120 Y180 I-60 J-60
...
```

Centrul sculei acoperă un arc de cerc în întregime din punctul P_1 în punctul P_1 şi un altul din punctul P_1 în punctul P_2.

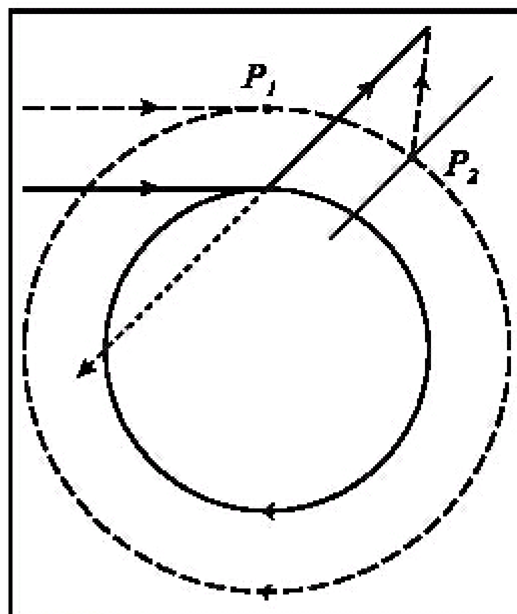


Fig.14.5.7-14

Doi sau mai mulți vectori de compensare pot fi produşi când se merge în jurul colțurilor ascuțite. Când punctele lor finale se găsesc aproape unele de celelalte, mişcarea nu va fi posibilă între două astfel de puncte. Când distanța dintre cei doi vectori este mai mică decât valoarea parametrului *DELTV* pe fiecare axă, vectorul arătat în figură va fi omis, şi traiectoria sculei va fi modificată în conformitate cu aceasta.

Notă: Când parametrul este prea mare (din diverse cauze) colțurile ascuțite cu unghiuri mici pot fi şterse la prelucrare.

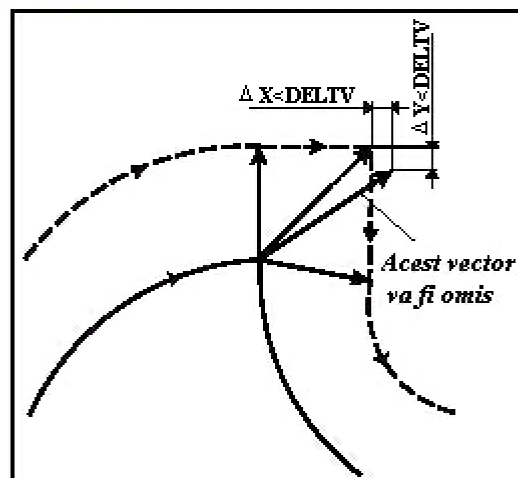


Fig. 14.5.7-15

14.5.8 Interferențe la compensarea sculei

Se poate întâmpla adesea în modul ofset ca traiectoria sculei să fie opusă față de cea programată. În aceste condiții, scula va tăia din piesă contrar cu intențiile programatorului. Se face referire la acest fenomen ca fiind interferență la compensarea sculei.

În cazul prezentat în figură, după ce s-au calculat punctele de intersecție, se va obține o traiectorie a sculei inversă față de cea programată în execuția interpolării N2. Aria haşurată indică faptul că scula taie din piesă.

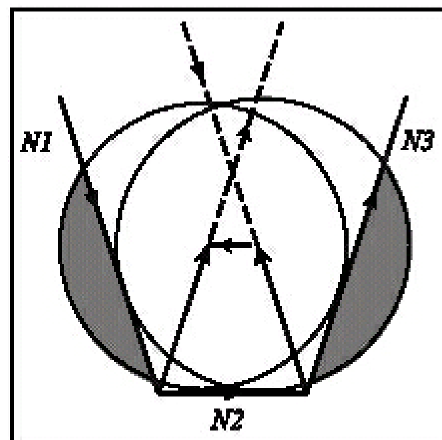


Fig. 14.5.8-1

Pentru a se evita aceasta, comanda numerică execută o verificare a interferenței atunci când parametrul *INTERFER* este impus 1. Acum comanda numerică va verifica dacă condiția $-90° \leq \varphi \leq +90°$ este îndeplinită pentru unghiul φ cuprins între deplasamentul programat şi cel compensat cu raza. Cu alte cuvinte comanda numerică va verifica dacă vectorul de deplasament compensat are sau nu o componentă opusă vectorului de deplasament programat.

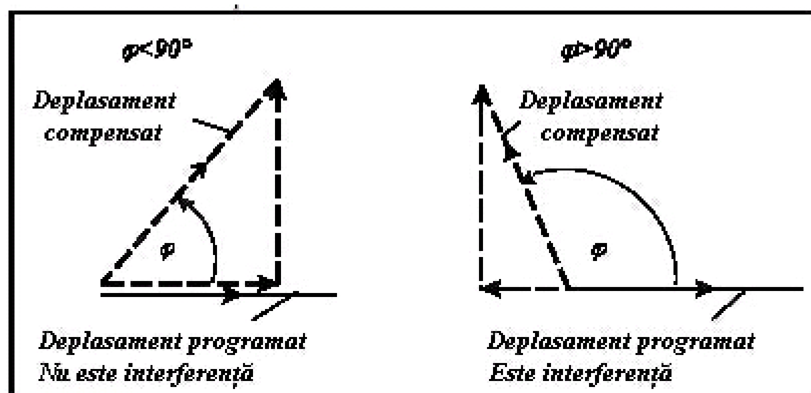


Fig. 14.5.8-2

Dacă parametrul *ANGAL* este impus 1, după verificarea unghiului comanda numerică va afişa un mesaj de eroare de interferență *3048 INTERFERENCE ALARM* cu un bloc înainte de apariția problemei.

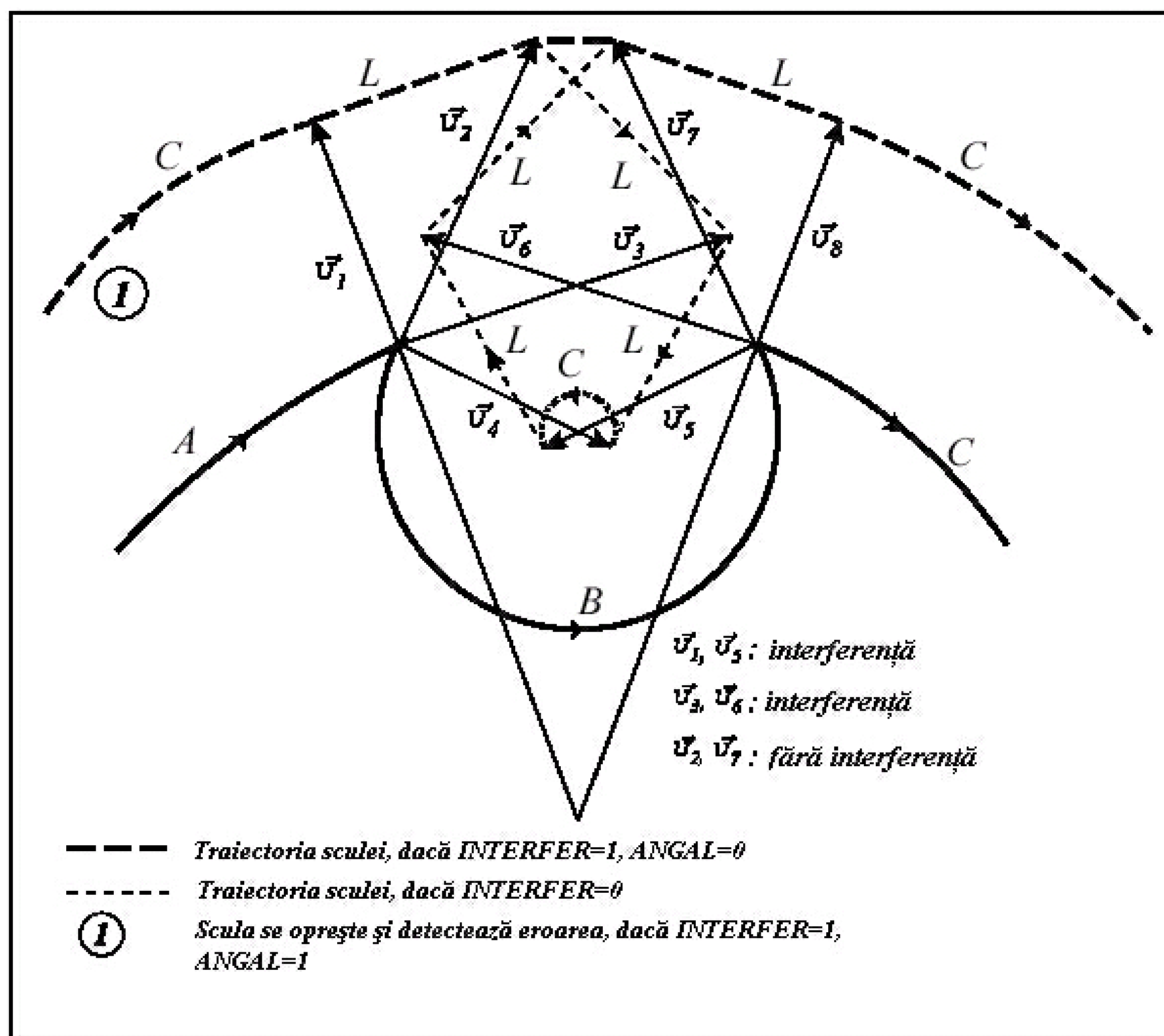


Fig. 14.5.8-3

Dacă parametrul *ANGAL* este impus 0, comanda numerică nu va afişa mesajul de eroare, dar va încerca în mod automat să corecteze conturul pentru a evita tăierea din piesă. În continuare se va descrie procedura de compensare.

Fiecare din blocurile A, B şi C sunt în modul ofset. Vectorii calculați între blocurile A şi B sunt V_1, V_2, V_3, V_4; vectorii de compensare între blocurile B şi C sunt V_5, V_6, V_7, V_8.

- V_4 şi V_5 vor fi ignorați dacă există o interferență între ei.
- V_3 şi V_6 vor fi ignorați dacă există o interferență între ei.
- V_2 şi V_7 vor fi ignorați dacă există o interferență între ei.
- V_1 şi V_8 nu pot fi omişi dacă există o interferență între ei, aşa că se afişează un mesaj de eroare.

Este evident din cele de mai sus că vectorii de compensare sunt împerecheați la începutul şi la sfârşitul punctelor de interpolare B şi vor fi ignorați în perechi. Dacă numărul de vectori de compensare pe o parte este 1 (sau se reduce la 1), vor fi omişi doar vectorii de pe cealaltă

parte. Procedura de omitere va fi executată atât timp cât interferenţa persistă. Primul vector de compensare din punctul de pornire al interpolării B şi ultimul vector din punctul final nu pot fi ignoraţi. Dacă, drept rezultat al omisiunii, este eliminată interferenţa, nu se va afişa un mesaj de eroare, dar în celelalte cazuri se va afişa mesajul *3048 INTERFERENCE ALARM*. Vectorii de compensare rămaşi după fiecare omisiune vor fi interconectaţi întotdeauna prin linii drepte – chiar dacă interpolarea B este circulară.
Este evident din exemplul de mai sus că execuţia interpolării A nu va fi începută până când nu s-a verificat pentru interferenţe interpolarea B. Pentru aceasta, şi blocul C trebuie introdus în bufer şi trebuie calculaţi şi vectorii de compensare pentru tranziţia B – C.

Mai jos se vor descrie câteva cazuri tipice de interferenţă:

Frezarea unei trepte mai mici decât raza sculei. Comanda numerică afişează mesajul de eroare *3048 INTERFERENCE ALARM* sau altfel va tăia din piesă.

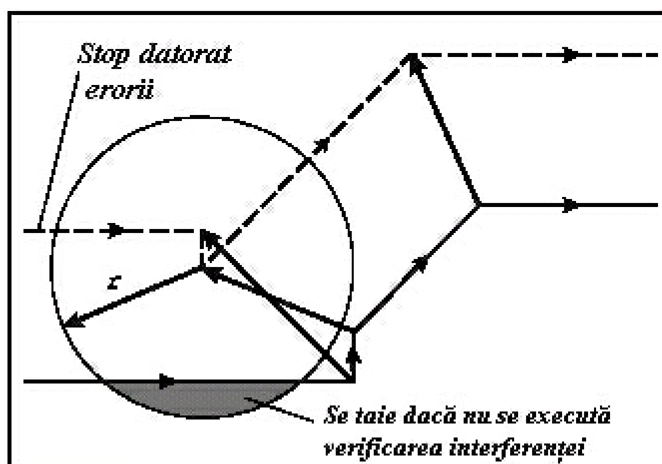


Fig.14.5.8-4

Prelucrarea unui colţ interior cu o rază mai mică decât raza sculei. Comanda numerică afişează mesajul de eroare *3048 INTERFERENCE ALARM* sau altfel va tăia mai mult din piesă.

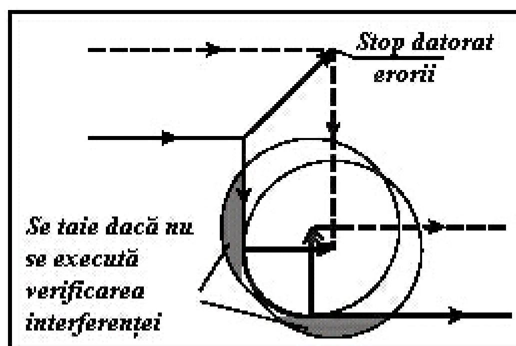


Fig.14.5.8-5

Frezarea unei trepte mai mici decât raza sculei de-a lungul unui arc. Dacă parametrul *ANGAL*=0, comanda numerică va şterge vectorul v_2 şi va interconecta vectorii v_1 şi v_3 cu o linie dreaptă pentru a evita tăierea din piesă. Dacă parametrul *ANGAL*=1, comanda numerică va afişa mesajul de eroare *3048 INTERFERENCE ALARM* şi se va opri la sfârşitul blocului anterior.

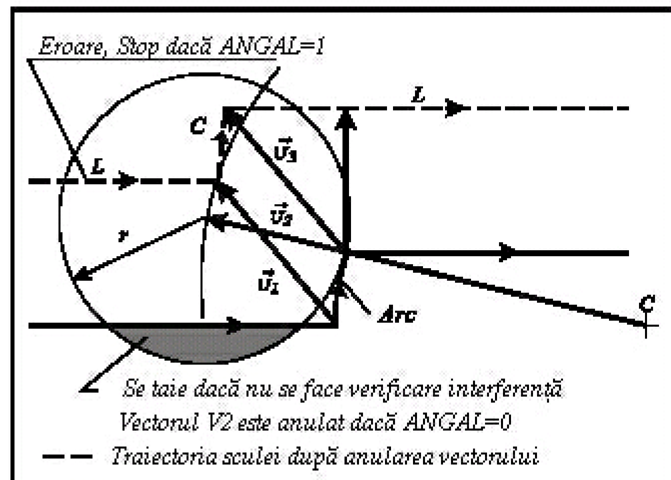


Fig. 14.5.8-6

Uneori scula nu va tăia efectiv din piesă, dar verificarea de interferenţă indică o eroare. Dacă se prelucrează un canal mai mic decât raza de compensare, nu se va tăia din piesă mai mult (vezi figura), totuşi comanda numerică va afişa mesajul de eroare *3048 INTERFERENCE ALARM* deoarece direcţia de deplasare în lungul traiectoriei de interpolare compensate B este opusă celei programate.

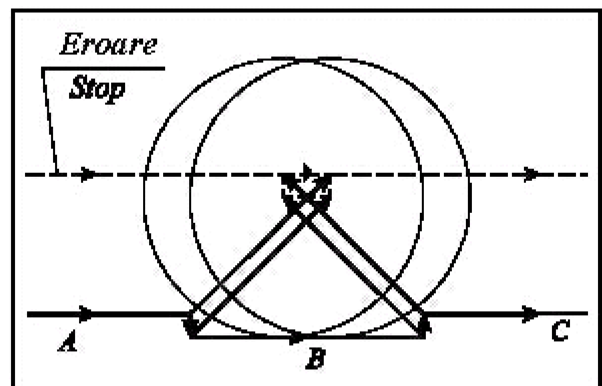


Fig. 14.5.8-7

În exemplul alăturat este afişată de asemenea o eroare de interferență deoarece deplasarea traiectoriei compensate în interpolarea B este opusă celei programate.

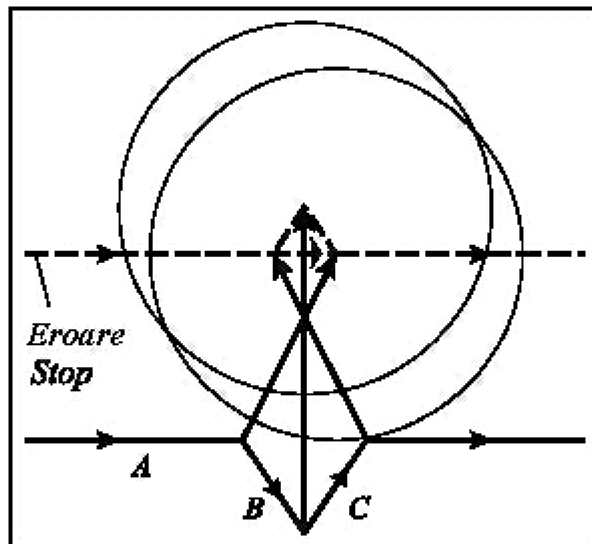


Fig. 14.5.8-8

14.6 Ofsetul tridimensional al sculei (G41, G42)

Compensarea 2D a razei sculei va produce ofsetul sculei în planul selectat prin comenzile G17, G18, G19. Aplicarea compensării tridimensionale a sculei face posibilă luarea în considerație a compensării sculei în trei dimensiuni.

14.6.1 Programarea ofsetului tridimensional al sculei (G40, G41, G42)

Comanda

G41 (G42) X_p Y_p Z_p I J K D (E)

va impune compensarea 3D a sculei.

X_p Y_p Z_p înseamnă axele X, Y, Z sau axe paralele cu acestea (dacă există).

Dacă nu se face referire la o anumită axă sunt luate în considerare în mod automat axele principale. De exemplu:

instrucțiunea G41 X I J K se referă la spațiul X Y Z

instrucțiunea G41 U V Z I J K se referă la spațiul U W Z

instrucțiunea G41 W I J K se referă la spațiul X Y W

Atunci când este impusă compensarea tridimensională a sculei, trebuie specificată fiecare dintre adresele I, J, K pentru că altfel comanda numerică va trece în starea de compensare bidimensională (2D) a razei.

Valorile specificate la adresele I, J, K sunt componentele vectorului de compensare tridimensional. Valorile componentelor sunt modale, adică fiecare dintre ele va rămâne activă până când se va face referire la o altă valoare I, J sau K.

Valoarea de compensare care urmează să fie aplicată poate fi apelată la adresa D.

Constanta dominantă a calculului compensării poate fi specificată la adresa E.

Comando

G40 sau

D00

va anula compensarea de ofset tridimensională.
Diferența între cele două comenzi este că D00 va şterge doar compensarea, păstrându-se starea G41 sau G42 neschimbată. Dacă după aceea se face referire la o nouă adresă D (alta decât zero), noua compensare a sculei va fi impusă în funcție de starea G41 sau G42. Dacă este folosită instrucțiunea G40, orice referire la adresa D nu va avea nici un efect până când nu se va programa G41 sau G42.
Calculul compensării poate fi impus (G41, G42) sau anulat (G40 sau D00) numai într-un bloc de mişcare liniară (G00 sau G01).
Stările G40, G41, G42 sunt modale. La punerea sub tensiune comanda numerică trece în starea G40.

14.6.2 Vectorul de ofset tridimensional
Comanda numerică va genera componentele vectorilor de compensare în modul următor:

$$v_x = \frac{I*r}{P}$$
$$v_y = \frac{J*r}{P}$$
$$v_z = \frac{K*r}{P}$$

unde r este valoarea de compensare apelată la adresa D,
P este constanta dominantă,
I, J, K sunt valorile specificate în program.
Valoarea constantei dominante este luată din parametrul *DOMCONST* doar în cazul în care nu este specificată o altă valoare în program la adresa E. Dacă valoarea constantei dominante este 0 şi la adresa E nu s-a specificat altă valoare, comanda numerică va calcula valoarea lui P din relația

$$P=\sqrt{I^2+J^2+K^2}$$

În funcție de direcțiile vectorilor de compensare specificați în fiecare bloc, comanda numerică va lua în considerație compensarea bloc după bloc. Astfel, în decursul unei prelucrări tridimensionale, sistemul CAM nu are nevoie să genereze traiectoria pentru o anumită sculă, ci trebuie calculate doar direcțiile vectoriale la punctele finale ale interpolărilor. Apoi programele generate în acest fel pot fi rulate şi folosind scule de lungimi diferite.

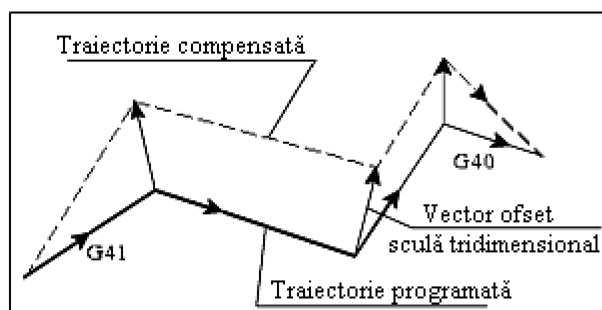


Fig. 14.6.2-1

Vectorul de compensare nu poate fi modificat într-o interpolare circulară, adică vectorii de compensare sunt identici la începutul şi la sfârşitul unei interpolări circulare.

Instrucţiunea G42 funcţionează în acelaşi fel ca G41 cu diferenţa că vectorul de compensare este calculat într-o direcţie opusă la G41:

$$v_x = -\frac{I*r}{P}$$
$$v_y = -\frac{J*r}{P}$$
$$v_z = -\frac{K*r}{P}$$

Trecerea din starea G41 în G42 sau invers se poate face numai într-un bloc de interpolare liniară.
Valorile anterioare vor fi modale dacă – fiind impusă compensarea tridimensională a sculei – într-o interpolare I, J şi K sunt toate omise.
Nu este posibilă impunerea simultană a compensării tridimensionale şi a compensării de rază bidimensionale.

15 Transformări speciale

15.1 Rotația sistemului de coordonate (G68, G69)

O formă programată poate fi rotită în planul selectat cu G17, G18, G19 prin folosirea comenzii

G68 p q R

Coordonatele centrului de rotație vor fi specificate la adresa p şi q. Comanda numerică va interpreta doar data scrisă la coordonatele p şi q ale planului selectat. Datele de coordonată p şi q sunt interpretate ca şi coordonate rectangulare chiar şi atunci când este impusă specificarea de coordonate polare. Folosind G90, G91 sau operatorul I, coordonatele p şi q ale centrului de rotație pot fi specificate ca date absolute sau incrementale.

În cazul în care una sau ambele valori pentru p şi q nu sunt introduse, va fi considerată ca centru de rotație poziția instantanee a axei.

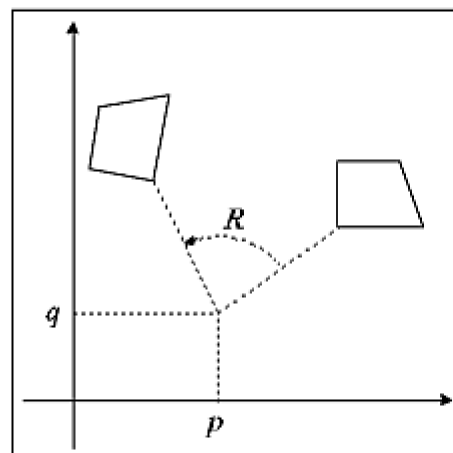


Fig. 15.1-1

Unghiul de rotație este specificat la adresa R. O valoare pozitivă sau negativă scrisă la adresa R reprezintă o direcție de rotație antiorară respectiv orară.

Valoarea lui R poate fi specificată cu 8 cifre zecimale. Acuratețea rotației poate fi selectată prin referirea la parametrul ANG.ACCU. Dacă valoarea sa este 0 sau 1, acuratețea de calcul a rotației va fi 0.001° respectiv 0.00001°.

Valoarea specificată pentru R poate fi absolută sau incrementală. Când unghiul de rotație este specificat ca dată incrementală, valoarea lui R va fi adăugată unghiului programat anterior.

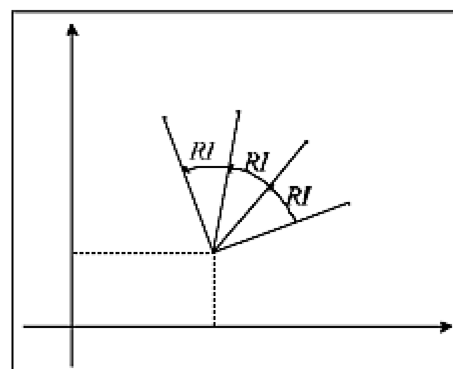


Fig. 15.1-2

Rotația poate fi anulată cu comanda

G69.

Coordonatele centrului de rotație şi unghiul vor fi şterse. Instrucțiunea poate acompania şi o altă comandă.

Exemplu:

```
N1 G17 G90 G0 X0 Y0
N2 G68 X90 Y60 R60
N3 G1 X60 Y20 F150
   (G91 X60 Y20 F150)
N4 G91 X80
N5 G3 Y60 R100
N6 G1 X-80
N7 Y-60
N8 G69 G90 X0 Y0
```

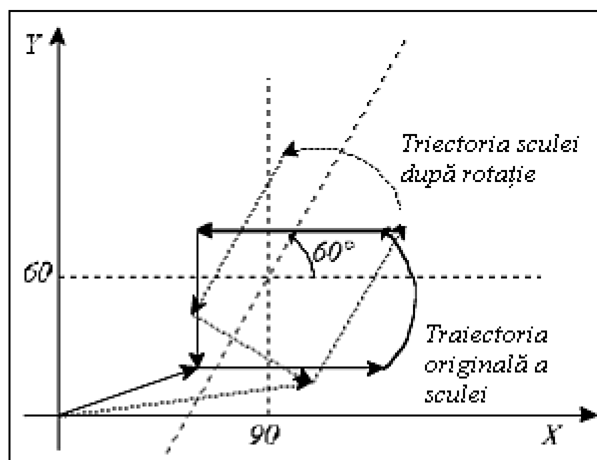


Fig. 15.1-3

15.2 Modificarea la scară (G50, G51)

Comanda

G51 v P

poate fi folosită pentru modificarea la scară a unei forme programate.

P1…P4: Puncte specificate în programul piesă
P'1…P'4: Puncte după modificare la scară
P0: Centrul de modificare la scară

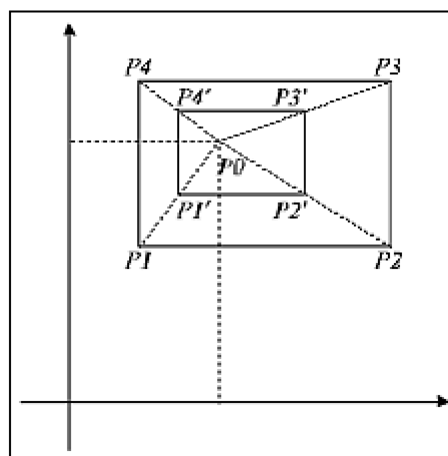


Fig. 15.2-1

Coordonatele centrului de modificare la scară pot fi introduse la coordonatele lui v. Adresele aplicabile sunt X, Y, Z, U, V, W. Datele pentru coordonata V introduse aici vor fi interpretate ca date rectangulare (carteziene), chiar şi atunci când este activă specificarea de date polare.
Folosind G90, G91 sau operatorul I, coordonatele v ale centrului de modificare la scară pot fi specificate ca date absolute sau incrementale.
În cazul în care adresele uneia sau a ambelor axe nu sunt valori asignate, va fi considerat centru de modificare la scară poziția instantanee a axelor.
Factorul de modificare la scară poate fi specificat la adresa P. Valoarea sa poate fi reprezentată prin 8 cifre zecimale; poziția punctului zecimal nu are importanță.
Modificarea la scară se poate anula cu comanda

G50

Exemplu:

```
N1 G90 G0 X0 Y0
N2 G51 X60 Y140 P0.5
N3 G1 X30 Y100 F150
(G91 X30 Y100 F150)
N4 G91 X100
N5 G3 Y60 R100
N6 G1 X-100
N7 Y-60
N8 G50 G90 X0 Y0
```

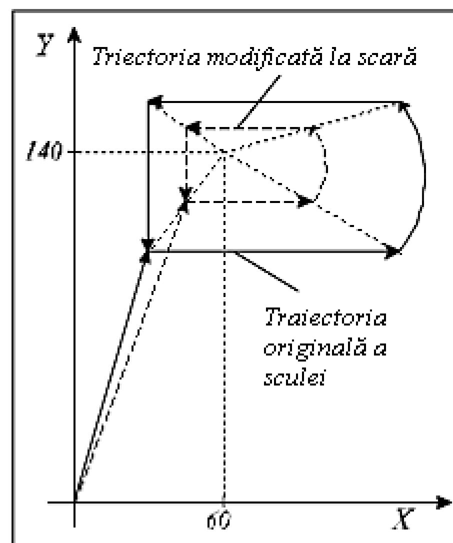


Fig. 15.2-2

15.3 Imagine în oglindă programabilă (G50.1, G51.1)

O formă programată poate fi proiectată ca o imagine în oglindă de-a lungul coordonatelor selectate în v prin comanda

G51.1 v

astfel încât coordonatele axei (sau axelor) imaginii în oglindă pot fi specificate prin v.
Coordonata v poate fi X, Y, Z, U, V, W, A, B, C.
Datele coordonatei v introduse aici sunt interpretate drept coordonate rectangulare chiar şi atunci când este activă specificarea de coordonate polare.
Folosind G90, G91 sau operatorul I, coordonatele v ale axelor imaginii în oglindă pot fi specificate ca date absolute sau incrementale.
Pe axa pentru a cărei adresă nu s-a introdus nici o valoare, nu va fi nici o imagine în oglindă.
Comanda

G50.1 v

va anula imaginea în oglindă pe axa (axele) specificate la v. Pentru coordonatele v se pot înscrie orice date arbitrare, efectul fiind doar înregistrarea acțiunii de anulare.
Atunci când se lansează această comandă nu trebuie să fie în lucru o comandă de rotație sau de modificare la scară. În caz contrar este afişat un mesaj de eroare *3000 MIRROR IMAGE IN G51, G68 (imagine în oglindă în G51, G68)*.
Atunci când se aplică o imagine în oglindă asupra unei axe din compunerea planului selectat:
- direcția de parcurgere a cercului este inversată automat (interschimbare G02 cu G03)
- unghiul de rotație este considerat cu semnificație inversă (G68).

Exemplu:

subprogram

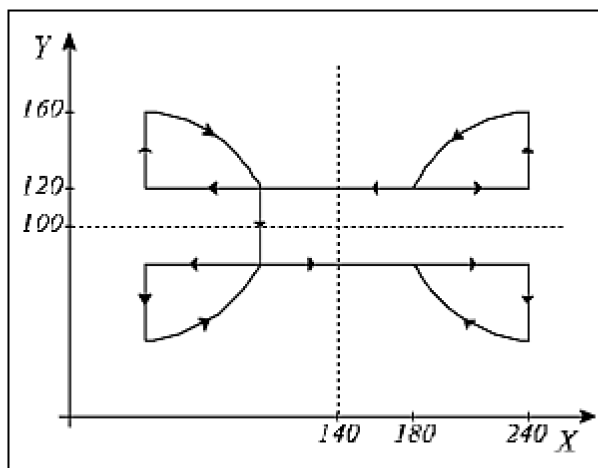


Fig. 15.3-1

```
O0101
N1 G90 G0 X180 Y120 F120
N2 G1 X240
N3 Y160
N4 G3 X180 Y120 R80
N5 M99
```

program principal

```
O0100
N1 G90 (specificare de coordonată absolută)
N2 M98 P101 (apel de subprogram)
N3 G51.1 X140 (imagine în oglindă aplicată unei axe paralele cu axa
                Y la coordonata X=140)
N4 M98 P101 (apel de subprogram)
N5 G51.1 Y100 (imagine în oglindă aplicată unei axe paralele cu axa
                X la coordonata Y=100)
N6 M98 P101 (apel de subprogram)
N7 G50.1 X0 (anulare imagine în oglindă pe axa paralelă cu Y)
N8 M98 P101 (apel de subprogram)
N9 G50.1 Y0 (anulare imagine în oglindă pe axa paralelă cu X)
```

15.4 Reguli de programare a transformărilor speciale

Instrucțiunile de rotație şi de modificare la scară G68 şi G51 pot fi introduse în orice ordine. Totuşi trebuie avut în vedere că – atunci când rotația este urmată de modificarea la scară - comanda de rotație va avea un efect asupra centrului de modificare la scară. Dacă, pe de altă parte, modificarea la scară este urmată de rotație, comanda de modificare la scară va avea un efect asupra coordonatelor centrului de rotație.
În continuare comenzile de activare şi dezactivare ale celor două proceduri trebuie să fie cuprinse unele în altele, ele nu trebuie să se intersecteze unele cu altele:

rotație-modificare la scară

```
N1 G90 G17 G0 X0 Y0
N2 G68 X80 Y40 R60
N3 G51 X130 Y70 P0.5
N4 X180 Y40
N5 G1 Y100 F200
N6 X80
N7 Y40
N8 X180
N9 G50
N10 G69 G0 X0 Y0
```

modificare la scară-rotație

```
N1 G90 G17 G0 X0 Y0
N2 G51 X130 Y70 P0.5
N3 G68 X80 Y40 R60
N4 X180 Y40
N5 G1 Y100 F200
N6 X80
N7 Y40
N8 X180
N9 G69
N10 G50 G0 X0 Y0
```

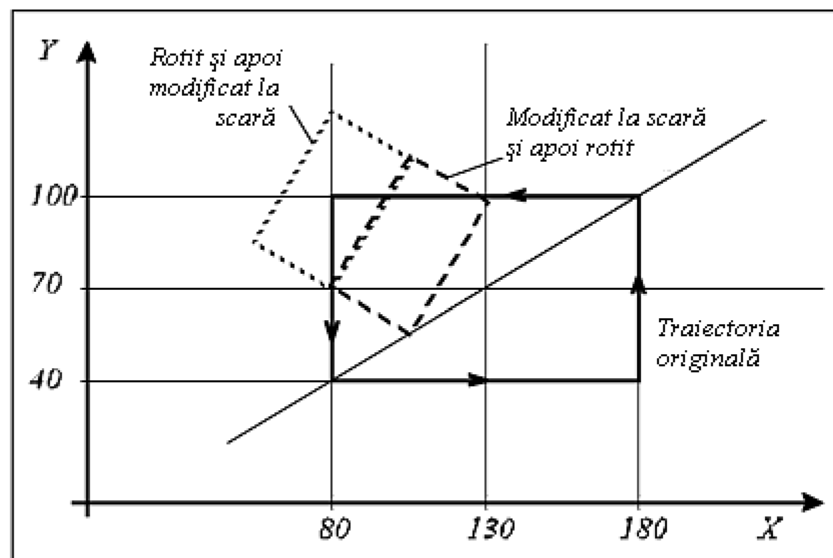


Fig. 15.4-1

Este evident din figura de mai sus faptul că ordinea de aplicare a diferitelor transformări are importanță.

Imaginea în oglindă programată este un caz diferit. Ea poate fi activată doar în stările G50 şi G69 adică în absența comenzilor de modificare la scară şi de rotație.
Pe de altă parte, în starea activă a comenzii de imagine în oglindă se pot activa atât comanda de modificare la scară cât şi cea de rotație.
Imaginea în oglindă nu se poate suprapune peste comenzile de modificare la scară şi de rotație.
În consecință trebuie anulate în ordinea corectă mai întâi rotația şi apoi modificarea la scară, iar apoi dezactivată imaginea în oglindă:

```
G51.1 …     (activare imagine în oglindă)
G51 …       (activare modificare la scară)
G68 …       (activare rotație)
…

G69 …       (dezactivare rotație)
G50 …       (dezactivare modificare la scară)
G50.1 …     (dezactivare imagine în oglindă)
```

16 Calcule geometrice automate

16.1 Programarea teşirii şi a rotunjirii colțurilor

Comanda numerică este capabilă să insereze automat teşirea sau rotunjirea între două blocuri care conțin interpolare liniară (G01) sau circulară (G02, G03).

O teşire a cărei lungime este egală cu valoarea specificată la adresa

,C

(virgulă şi C) este inserată între punctul final al blocului care conține adresa **,C** şi punctul de început al blocului următor.

Exemplu:

```
N1 G1 G91 X30 ,C10
N2 X10 Y40
```

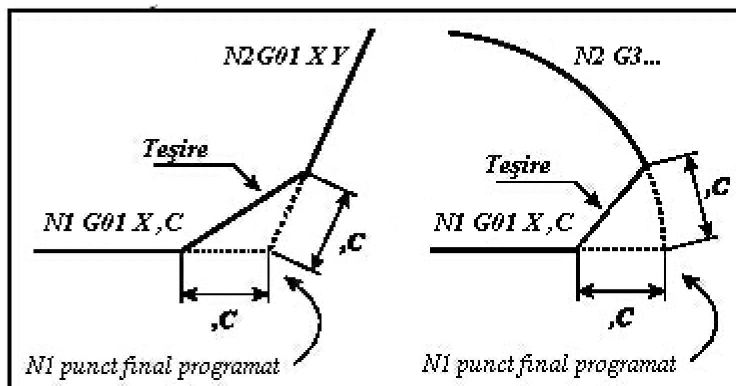


Fig. 16.1-1

Valoarea specificată la adresa **,C** arată distanța dintre punctul de început/sfârşit al teşiturii şi intersecția presupusă a celor două blocuri succesive. Se poate insera o teşitură şi între două cercuri sau între un cerc şi o linie dreaptă. În acest caz valoarea **,C** este lungimea coardei desenate din intersecție.

O rotunjire, a cărei rază corespunde valorii date la adresa

,R

(virgulă şi R) este inserată între punctul final al blocului care conține adresa **,R** şi punctul de început al blocului următor.

Exemplu:

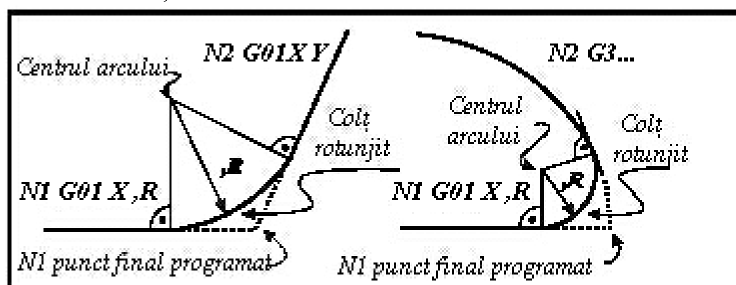


Fig. 16.1-2

```
N1 G91 G01 X30 ,R8
N2 G03 X-30 Y30 R30
```

Se inserează un arc de rază ,R între cele două blocuri astfel încât cercul să se racordeze la ambele elemente de traiectorie.

Comanda conținând o teşitură sau o rotunjire a colțului poate fi scrisă şi la sfârşitul mai multor blocuri succesive după cum se arată în exemplul următor:

```
…
G1 Y40 ,C10
X60 ,R22
G3 X20 Y80 R40 ,C10
G1 Y110
…
```

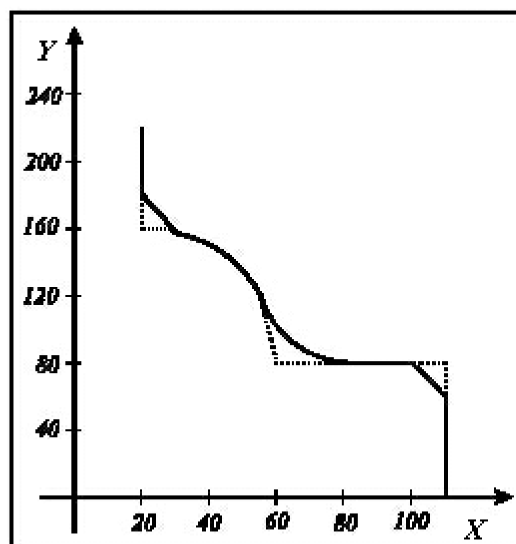


Fig. 16.1-3

Note:
- Teşirea sau rotunjirea pot fi programate doar între coordonatele planului selectat (G17, G18, G19), în caz contrar se afişează mesajul de eroare *3081 DEFINITION ERROR ,C ,R (eroare de definire pentru ,C ,R)*.
- Teşirea sau rotunjirea colțului pot fi aplicate doar între blocurile G1, G2 sau G3, în caz contrar se afişează mesajul de eroare *3081 DEFINITION ERROR ,C ,R (eroare de definire pentru ,C ,R)*.
- În cazul în care lungimea teşiturii sau raza rotunjirii sunt aşa de mari încât nu pot fi inserate la blocurile programate, se afişează mesajul de eroare *3084 ,C ,R TOO HIGH (,C ,R prea mari)*.
- Dacă sunt programate în acelaşi bloc ,C şi ,R se afişează mesajul de eroare *3017 ,C AND ,R IN ONE BLOCK (,C şi ,R în acelaşi bloc)*.
- În modul bloc singular comanda numerică se opreşte şi înregistrează starea STOP după executarea teşiturii sau rotunjirii colțului.

16.2 Specificarea liniei drepte prin unghi

Linia dreaptă poate fi specificată în planul determinat de comenzile G17, G18, G19 cu ajutorul unei coordonate a planului selectat şi a unghiului dat la adresa ,A**.**

$$G17 \begin{Bmatrix} G00 \\ \hline G01 \end{Bmatrix} \begin{Bmatrix} X_p \ ,A \\ \hline Y_p \ ,A \end{Bmatrix} q \ F$$

$$G18 \begin{Bmatrix} G00 \\ \hline G01 \end{Bmatrix} \begin{Bmatrix} Z_p \ ,A \\ \hline X_p \ ,A \end{Bmatrix} q \ F$$

$$G19 \begin{Bmatrix} G00 \\ \hline G01 \end{Bmatrix} \begin{Bmatrix} Y_p \ ,A \\ \hline Z_p \ ,A \end{Bmatrix} q \ F$$

În formulele de mai sus X_p, Y_p, Z_p indică axele X, Y, Z sau pe cele paralele cu acestea, în timp ce q reprezintă o axă opțională în afara planului selectat. Specificațiile de la adresa ,A pot fi folosite şi după codurile G0 şi G1. Unghiul este măsurat de la prima axă a planului selectat şi direcția pozitivă este antiorară. Valoarea ,A poate fi atât pozitivă cât şi negativă, mai mare de 360° sau mai mică de –360° .

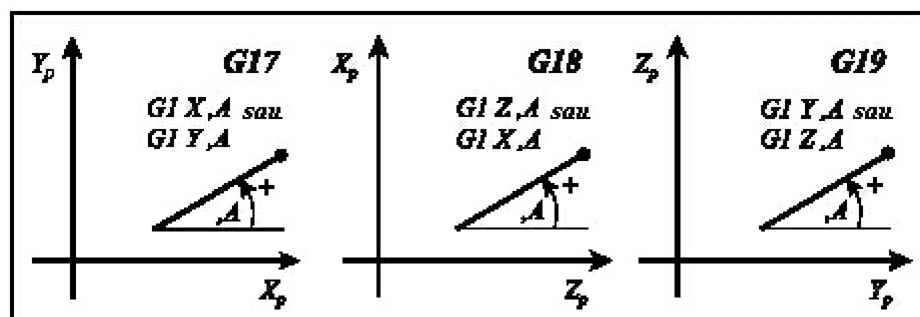


Fig. 16.2-1

Exemplu:

```
G17 G90 G0 X57.735 Y0 ...
G1 G91...
X100 ,A30 (această specificaţie
           este echivalentă cu
           X100 Y57.735 unde
           7.735=100*tg30°)
Y100 ,A120 (această specificaţie
            este echivalentă cu
            X-57.735 Y100 unde
            -57.735=100/tg120°)
X-100 ,A210 (această specificaţie
             este echivalentă cu
             X-100 Y-57.735 unde
            -57.735=-100*tg30°)
Y-100 ,A300 (această specificaţie
             este echivalentă cu
             X57.735 Y-100 unde
             57.735=-100/tg120°)
```

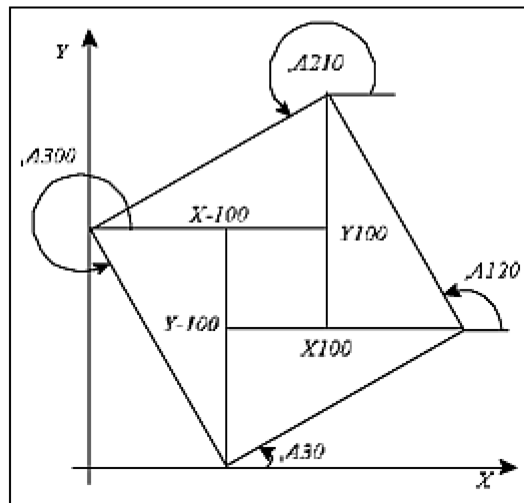


Fig. 16.2-2

Note.

- Se poate defini într-un bloc o linie dreaptă prin unghi, împreună cu o teşitură sau o rotunjire de colţ. Exemplu:

```
X100 ,A30 ,C5
Y100 ,A120 ,R10
X-100 A210
```

- Specificarea unghiului la adresa ,A se poate aplica şi la ciclurile de găurire. În acest caz este recunoscut în cursul executării poziţionării în planul selectat.

De exemplu blocul:

```
G81 G91 X100 ,A30 R-2 Z-25
```

este echivalent cu blocul următor:

```
G81 G91 X100 Y57.735 R-2 Z-25
```

16.3 Calculul intersecției în planul selectat

Calculul intersecției discutat în continuare este executat de către comanda numerică doar când ***este activă compensarea razei sculei (modul ofset G41 sauG42).*** Dacă în programul piesă nu este nevoie de compensarea razei sculei, se activează compensarea şi se foloseşte un ofset D00:

Cu compensarea razei sculei:	Fără compensarea razei sculei:
`G41(sau G42) ...Dnn`	`G41(sau G42) ...D00`
`...`	`...`
`calculele intersecției`	`calculele intersecției`
`...`	`...`
`G40`	`G40`

16.3.1 Intersecția liniar – liniar

Dacă din două blocuri succesive de interpolare liniară, al doilea bloc este specificat astfel încât sunt specificate coordonatele punctului final în planul selectat şi de asemenea unghiul său, comanda numerică calculează intersecția liniei drepte la care se face referire în primul bloc şi linia dreaptă specificată în cel de al doilea. Linia dreaptă specificată în cel de al doilea bloc este determinată în totalitate. Intersecția calculată este punctul final al primului bloc, şi de asemeni punctul de pornire al celui de al doilea.

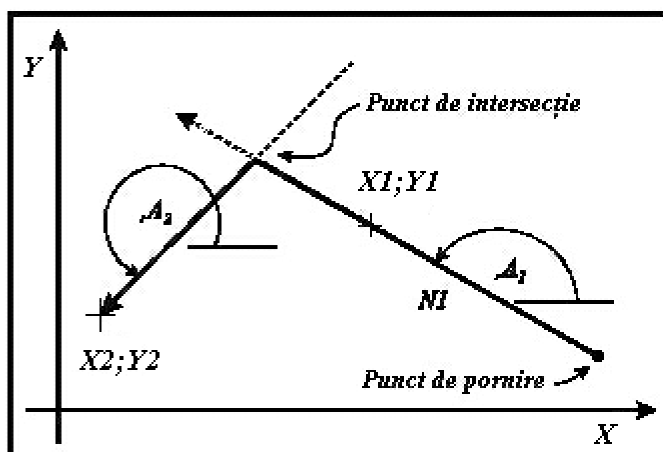


Fig. 16.3.1-1

`G17 G41 (G42)`	`G18 G41 (G42)`	`G19 G41 (G42)`
`N1 G1 ,A1 sau`	`N1 G1 ,A1 sau`	`N1 G1 ,A1 sau`
`X1 Y1`	`X1 Z1`	`Y1 Z1`
`N2 G1 G90` **`X2 Y2 ,A2`**	`N2 G1 G90` **`X2 Z2 ,A2`**	`N2 G1 G90` **`Y2 Z2 ,A2`**

Intersecția este calculată întotdeauna în planul selectat de G17, G18, G19. Primul bloc (N1) este specificat fie cu ajutorul unghiului său (,A1), în acest caz este dusă o linie dreaptă în unghiul respectiv din punctul inițial până în punctul de intersecție, sau printr-un poziție opțională, alta decât punctul inițial al liniei drepte (X1, Y1; X1, Z1; sau Y1, Z1). În acest caz intersecția este calculată cu linia dreaptă care trece prin ambele puncte. Coordonatele date în al doilea bloc (N2) sunt interpretate întotdeauna de către comanda numerică ca date absolute (G90).

De exemplu:

```
(G17) G90 G41 D0...
G0 X90 Y10
N10 G1 ,A150
N20 X10 Y20 ,A225
G0 X0 Y20
...
```

Blocul N10 poate fi de asemenea dat cu coordonatele unui punct al liniei drepte:

```
G17 G90 G41 D0...
G0 X90 Y10
N10 G1 X50 Y33.094
N20 X10 Y20 ,A225
G0 X0 Y20
...
```

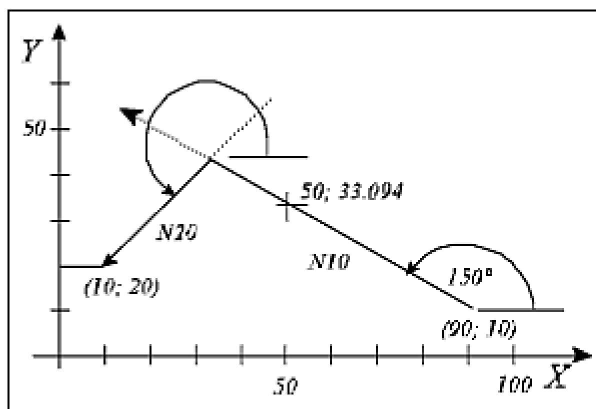


Fig. 16.3.1-2

De notat că în acest caz coordonata X, Y (X50 Y33.094) dată în blocul N10 nu e recunoscută de comanda numerică drept punct final, ci ca punct de tranziție care uneşte linia dreaptă cu punctul inițial.

Calculul intersecției poate fi combinat de asemenea cu specificare de teşire sau rotunjire de colț. Exemplu:

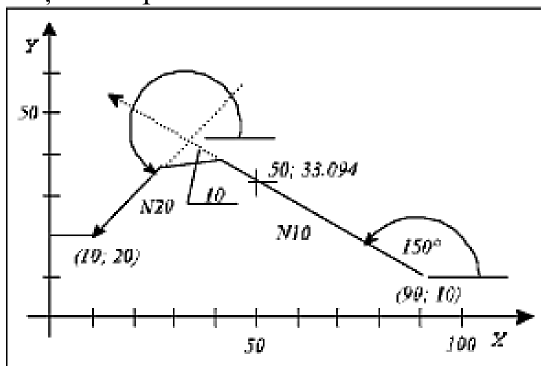


Fig. 16.3.1-3

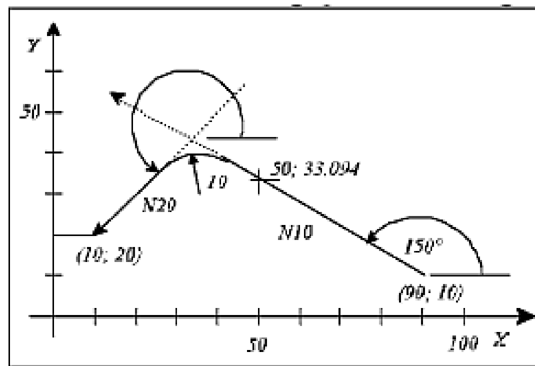


Fig. 16.3.1-4

```
G17 G90 G41 D0...
G0 X90 Y10
N10 G1 X50 Y33.094 ,C10
N20 X10 Y20 ,A225
G0 X0 Y20
...
```

```
G17 G90 G41 D0...
G0 X90 Y10
N10 G1 X50 Y33.094 ,R10
N20 X10 Y20 ,A225
G0 X0 Y20
...
```

În exemplele de mai sus mărimea teşirii este măsurată de la intersecția calculată, iar rotunjirea este inserată în intersecția calculată.

16.3.2 Intersecția liniar – circular

Dacă se dă un bloc circular după un bloc liniar, astfel că sunt specificate coordonatele poziției finale şi ale centrului şi razei cercului, adică cercul este determinat, atunci comanda numerică calculează intersecția între linia dreaptă şi cerc. Intersecția calculată este punctul final al primului bloc, ca şi punctul de început al celui de al doilea.

G17 G41 (G42)	G18 G41 (G42)	G19 G41 (G42)
N1 G1 ,A sau	N1 G1 ,A sau	N1 G1 ,A sau
X1 Y1	X1 Z1	Y1 Z1
N2 G2 (G3) G90 X2 Y2 I J R Q	N2 G2 (G3) G90 X2 Z2 I K R Q	N2 G2 (G3) G90 Y2 Z2 I K R Q

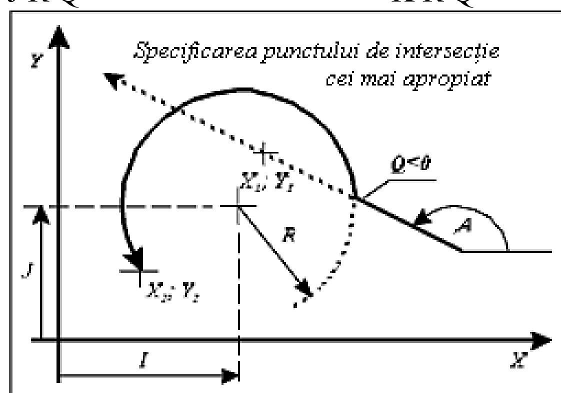


Fig. 16.3.2-1

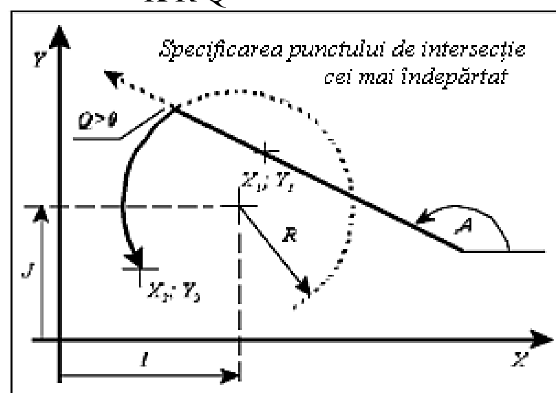


Fig. 16.3.2-2

Intersecția se calculează întotdeauna în planul selectat cu G17, G18, G19. Primul bloc (N1) este specificat fie prin unghiul său (,A), în acest caz este dusă o linie dreaptă din punctul inițial în punctul de intersecție sub unghiul corespunzător, sau prin specificarea unui punct opțional, altul decât punctul inițial al liniei drepte (X_1, Y_1; X_1, Z_1; sau Y_1, Z_1). În acest caz este calculată intersecția cu linia dreaptă care trece prin ambele puncte. Coordonatele date în al doilea bloc (N2) şi de asemeni **coordonatele I, J, K** care definesc **centrul arcului**, sunt interpretate întotdeauna de către comanda numerică ca date **absolute** (G90). Se poate specifica la adresa Q care dintre cele două intersecții rezultante să fie calculată.
Dacă valoarea adresei este mai mică decât zero (Q<0), este calculată cea mai apropiată intersecție în direcția liniei drepte, iar dacă valoarea adresei este mai mare decât zero (Q>0), este calculată cea mai îndepărtată intersecție în direcția liniei drepte. Direcția de mişcare de-a lungul liniei drepte este determinată de către unghi.

Să examinăm următorul exemplu:

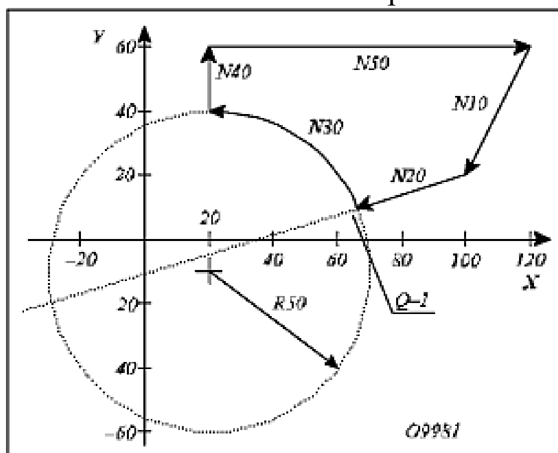


Fig. 16.3.2-3

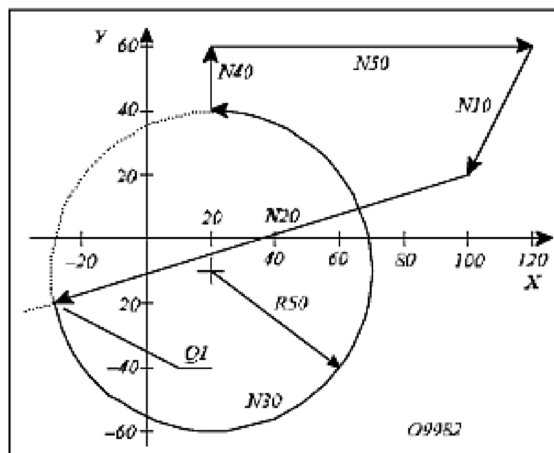


Fig. 16.3.2-4

```
%O9981
N10 G17 G42 G0 X100 Y20 D0 S200 M3
N20 G1 X-30 Y-20
N30 G3 X20 Y40 I20 J-10 R50 Q-1
N40 G40 G0 Y60
N50 X120
N60 M30
%
```

```
%O9982
N10 G17 G42 G0 X100 Y20 D0 S200 M3
N20 G1 X-30 Y-20
N30 G3 X20 Y40 I20 J-10 R50 Q1
N40 G40 G0 Y60
N50 X120
N60 M30
%
```

Blocul circular N30 G3 este determinat, fiind specificate atât coordonatele centrului (I20 J-10 în valoare absolută) cât şi raza cercului (R50), comanda numerică va calcula intersecţia liniei drepte date în blocul N20 cu cercul dat în blocul N30. În programul O9981 se calculează cea mai apropiată intersecţie în direcţia liniei drepte deoarece este dat Q-1 în blocul circular N30.

Calcului intersecţiei liniar – circular poate fi combinat cu specificarea teşirii sau rotunjirii colţului. Exemplu:

```
%O9983
N10 G17 G42 G0 X100 Y20 D0 S200 M3
N20 G1 X-30 Y-20 ,R15
N30 G3 X20 Y40 I20 J-10 R50 Q-1
N40 G40 G0 X60
N50 X120
N60 M30
%
```

Comanda numerică calculează intersecţia blocurilor N20 şi N30 şi introduce o rotunjire a colţului de 15 mm ca efect al **,R15** dat în blocul N20.

16.3.3 Intersecția circular - liniar

Dacă se dă un bloc liniar după un bloc circular astfel că linia dreaptă este total definită, adică sunt specificate atât punctul final cât şi unghiul, atunci comanda numerică calculează intersecția dintre cerc şi linia dreaptă. Intersecția calculată este punctul final al primului bloc, şi punctul de pornire al celui de al doilea.

G17 G41 (G42)	G18 G41 (G42)	G19 G41 (G42)
N1 G2 (G3) X1 Y1 I J	N1 G2 (G3) X1 Z1 I K	N1 G2 (G3) Y1 Z1 J K
sau R	sau R	sau R
N2 G1 G90 X2 Y2 ,A Q	N2 G1 G90 X2 Z2 ,A Q	N2 G1 G90 Y2 Z2 ,A Q

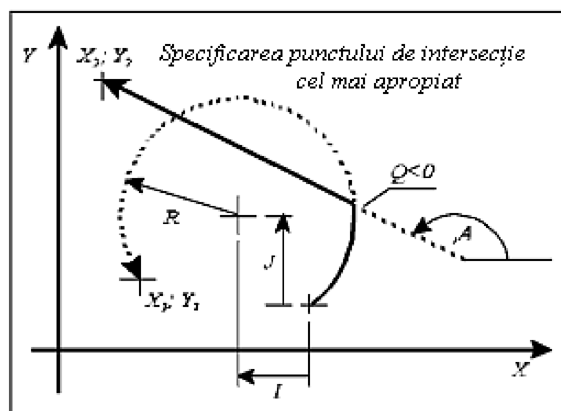


Fig. 16.3.3-1

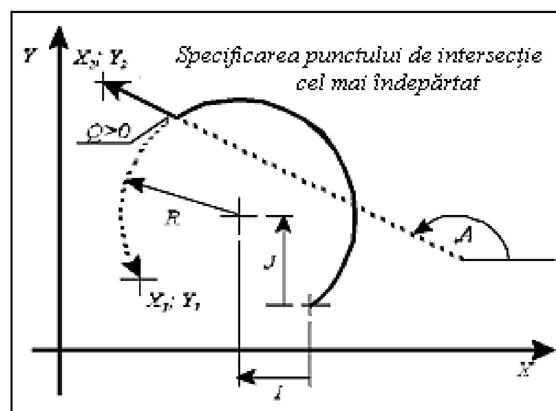


Fig. 16.3.3-2

Intersecția se calculează întotdeauna în planul selectat cu G17, G18, G19. În primul bloc (N1) este specificat cercul fie printr-un punct opțional (X_1, Y_1; X_1, Z_1; sau Y_1, Z_1) şi coordonatele punctului de centru (I J; I K; sau J K) fie prin coordonatele punctului de centru şi a razei (R). În al doilea bloc (N2) este determinată linia dreaptă, adică sunt date coordonatele punctului final (X_2, Y_2; X_2, Z_2; sau Y_2, Z_2) şi unghiul (,A) liniei drepte. Coordonatele punctului final al liniei drepte sunt interpretate întotdeauna de către comanda numerică ca date **absolute** (G90). La adresa **,A** trebuie specificat **unghiul vectorului** liniei drepte îndreptat de la **intersecția** rezultantă către **punctul final**, pentru că altfel pot apare mişcări inverse față de cele programate. Se poate specifica la adresa Q care dintre cele două intersecții rezultante să fie calculată.

Dacă valoarea adresei este mai mică decât zero (Q<0, de exemplu Q-1), este calculată cea mai apropiată intersecție în direcția liniei drepte, iar dacă valoarea adresei este mai mare decât zero (Q>0, de exemplu Q1), este calculată cea mai îndepărtată intersecție în direcția liniei drepte. Direcția de mişcare de-a lungul liniei drepte este determinată de către unghi.

Să examinăm următorul exemplu:

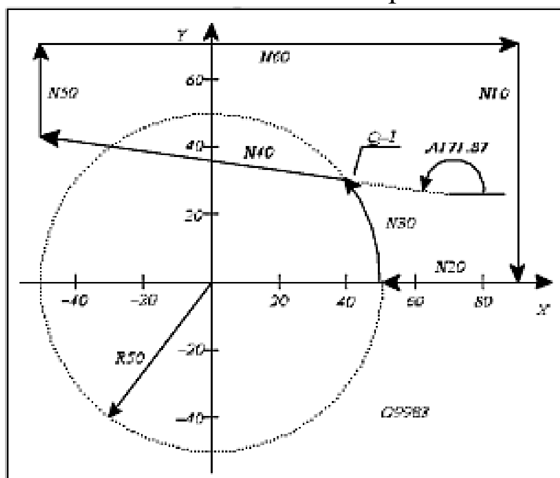


Fig. 16.3.3-3

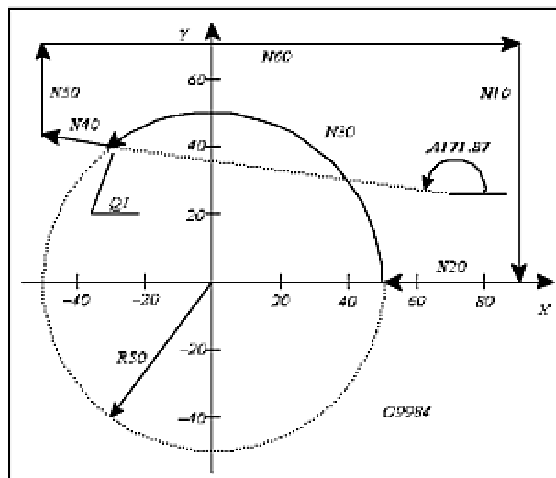


Fig. 16.3.3-4

```
%O9983
N10 G17 G0 X90 Y0 M3 S200
N20 G42 G1 X50 D0
N30 G3 X-50 Y0 R50
N40 G1 X-50 Y42.857 ,A171.87 Q-1
N50 G40 G0 Y70
N60 X90
N70 M30
%
```

```
%O9984
N10 G17 G0 X90 Y0 M3 S200
N20 G42 G1 X50 D0
N30 G3 X-50 Y0 R50
N40 G1 X-50 Y42.857 ,A171.87 Q1
N50 G40 G0 Y70
N60 X90
N70 M30
%
```

Blocul liniar N40 este complet definit, deoarece sunt specificate atât coordonatele punctului final (X-50 Y42.875) cât şi unghiul (,A171.87). Prin urmare coordonatele X-50 Y0 ale cercului programat în blocul anterior N30 nu sunt referite ca şi coordonate ale punctului final, ci doar ca punct care este intersectat de către cerc şi punctul final este intersecţia calculată. În programul nr. O9983 se calculează cea mai apropiată intersecţie în direcţia liniei drepte (Q-1), iar în O9984 este specificată cea mai îndepărtată intersecţie (Q1)

Calcului intersecţiei liniar – circular poate fi combinat cu specificarea teşirii sau rotunjirii colţului. Exemplu:

```
%O9983
N10 G17 G0 X90 Y0 M3 S200
N20 G42 G1 X50 D0
N30 G3 X-50 Y0 R50 ,R15
N40 G1 X-50 Y42.875 ,A171.87 Q-1
N50 G40 G0 Y70
N60 X90
N70 M30
%
```

În exemplu este programată o rotunjire de 15 mm în blocul N30 (,R15). Comanda numerică calculează intersecţia blocurilor N30 şi N40 şi introduce rotunjirea programată la conturul rezultant.

16.3.4 Intersecția circular - circular

Dacă sunt specificate două blocuri circulare succesive astfel ca sunt date punctul final, coordonatele centrului şi raza pentru cel de al doilea bloc, adică acesta este determinat, comanda numerică va calcula intersecția dintre cele două cercuri. Intersecția calculată este punctul final al primului bloc şi de asemeni punctul de pornire al celui de al doilea.

G17 G41 (G42)	G18 G41 (G42)	G19 G41 (G42)
N1 G2 (G3) X1 Y1 I1 J1	N1 G2 (G3) X1 Z1 I1 K1	N1 G2 (G3) Y1 Z1 J1 K1
sau X1 Y1 R1	sau X1 Z1 R1	sau Y1 Z1 R1
N2 G2 (G3) G90 X2 Y2	N2 G2 (G3) G90 X2 Z2	N2 G2 (G3) G90 Y2 Z2
I2 J2 R2 Q	I2 K2 R2 Q	J2 K2 R2 Q

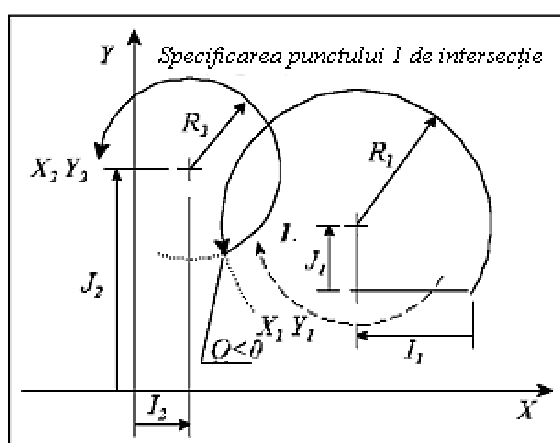


Fig. 16.3.4-1

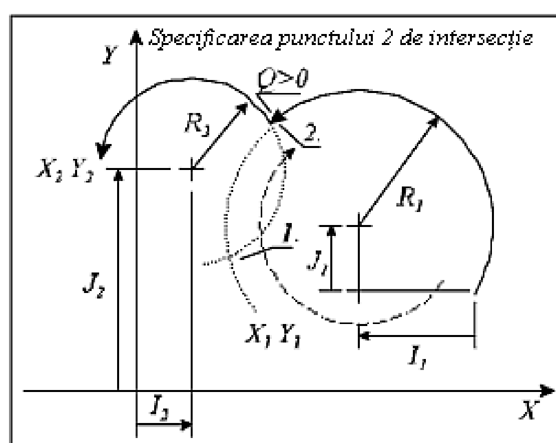


Fig. 16.3.4-2

Intersecția este calculată întotdeauna în planul selectat de G17, G18, G19. Primul bloc (N1) este specificat fie prin coordonatele centrului (I_1 J_1; I_1 K_1; sau J_1 K_1) fie cu raza cercului (R_1). În acest bloc interpretarea coordonatei centrului corespunde specificării implicite a cercului, adică este distanța relativă de la punctul de pornire. Coordonatele date în cel de al doilea bloc (N2), adică **coordonatele I, J, K** care definesc **centrul cercului**, sunt interpretate întotdeauna de către comanda numerică ca date **absolute** (G90). Se poate specifica la adresa Q care dintre cele două intersecții rezultante să fie calculată. Dacă valoarea adresei este mai mică decât zero (Q<0, de exemplu Q-1), este calculată prima intersecție, iar dacă valoarea adresei este mai mare decât zero (Q>0, de exemplu Q1), este calculată cea de a doua. **Prima intersecție este aceea care este intersectată prima mergând în sens orar (independent de direcția programată G2, G3).**

Să examinăm următorul exemplu:

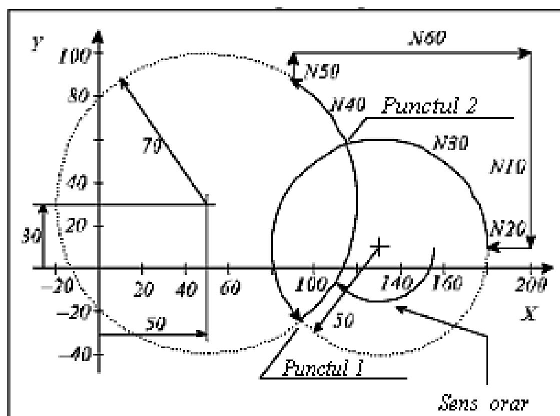


Fig. 16.3.4-3

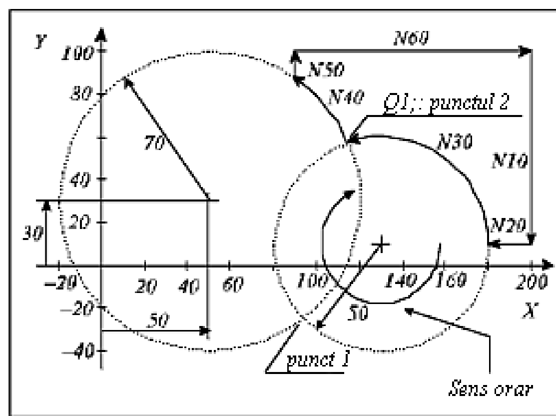


Fig. 16.3.4-4

```
%O9985
N10 G17 G54 G0 X200 Y0 M3 S200
N20 G42 G1 X180 D1
N30 G3 X130 Y-40 R-50
N40 X90 Y87.446 I50 J50 R70 Q-1
N50 G40 G0 Y100
N60 X200
N70 M30
%
```

```
%O9986
N10 G17 G54 G0 X200 Y10 M3 S200
N20 G42 G1 X180 D1
N30 G3 X130 Y-40 R-50
N40 X90 Y87.446 I50 J30 R70 Q1
N50 G40 G0 Y100
N60 X200
N70 M30
%
```

Blocul circular N40 este definit, deoarece sunt specificate atât coordonatele centrului (I50 J30 în valoare absolută) cât şi raza cercului (R70). Prin urmare coordonatele X-80 Z130 ale cercului programat în blocul anterior N30, nu sunt referite ca şi coordonate ale punctului final, ci doar ca punct care este intersectat de către cerc şi punctul final este intersecția calculată. În programul nr. O9985 este specificată cea mai apropiată intersecție în sens orar (Q-1), iar în O9986 este specificată cea mai îndepărtată intersecție (Q1)

Calculul intersecției circular – circular poate fi combinat cu specificarea teşirii sau rotunjirii colțului. Exemplu:

```
%O9986
N10 G17 G54 G0 X200 Y10 M3 S200
N20 G42 G1 X180 D1
N30 G3 X130 Y-40 R-50 ,R20
N40 X90 Y87.446 I50 J30 R70 Q1
N50 G40 G0 Y100
N60 X200
N70 M30
%
```

În exemplu este programată o rotunjire de 20 mm în blocul N30 (,R20). Comanda numerică calculează intersecția blocurilor N30 şi N40 şi introduce rotunjirea programată la conturul rezultant.

16.3.5 Înlănțuirea calculărilor intersecțiilor

Blocurile de calcul al intersecțiilor pot fi înlănțuite, adică pot fi selectate mai multe blocuri succesive pentru calculul intersecțiilor. Comanda numerică va calcula intersecția până când sunt găsite linii drepte sau cercuri determinate.
Să examinăm exemplul de mai jos:

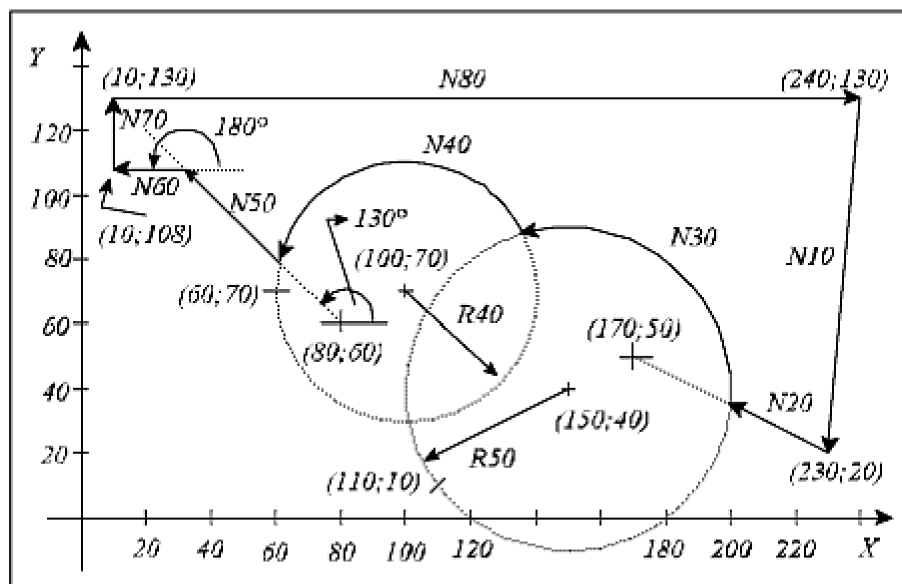


Fig. 16.3.5-1

```
%O9984
N10 G17 G54 G0 G42 X230 Y20 D1 F300 S500 M3
N20 G1 X170 Y50
N30 G3 X110 Y10 I150 J40 R50 Q-1
N40 X60 Y70 I100 J70 R40 Q1
N50 G1 X80 Y60 ,A135 Q1
N60 X10 Y108 ,A180
N70 G40 G0 Y130
N80 X240
N90 M30
%
```

În exemplul de mai sus, blocurile N30, N40, N50, N60 sunt determinate. Blocul liniar N20 nu este dus în punctul său final programat (X170 Y50) deoarece blocul circular N30 este definit, adică adresele I, J, R sunt completate şi intersecția care trebuie căutată este dată la adresa Q. Nici blocul N30 nu este dus în punctul său final programat (X110 Y10) deoarece blocul circular N40 este de asemenea definit. Ultimul bloc determinat în program este blocul liniar N60. Deoarece următorul bloc liniar N70 nu este definit, coordonatele X10 Y108 programate în blocul N60 nu sunt referite ca punct de intersecție al liniei drepte ci ca punct de coordonate final al blocului N60.
În general este adevărat că punctele de coordonate al blocurilor liniare şi circulare determinate în planul selectat sunt referite de către comanda numerică ca puncte de coordonate finale doar dacă nu sunt urmate de un bloc definit.

17 Cicluri memorate pentru găurire

Un ciclu de găurire poate fi descompus în următoarele operații:

Operația 1 : Poziționare în planul selectat
Operația 2 : Operare după poziționare
Operația 3 : Deplasare cu avans rapid în punctul R
Operația 4 : Operare în punctul R
Operația 5 : Găurire
Operația 6 : Operare în fundul găurii
Operația 7 : Retragere în punctul R
Operația 8 : Operare la punctul R
Operația 9 : Retragere cu avans rapid în punctul inițial
Operația 10: Operare la punctul inițial

Punctul R, punct de apropiere. – Scula se apropie de piesă cu avans rapid până în acest punct .

Punctul inițial. – Poziția de pe axa de găurire în care trebuie să se ajungă înainte de pornirea ciclului.

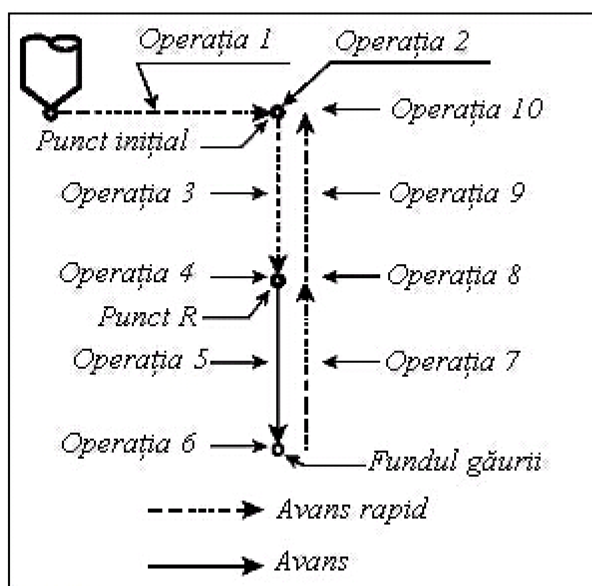


Fig. 17-1

Operațiile de mai sus dau o vedere generală asupra ciclului de găurire; în anumite circumstanțe anumite operații se pot omite.
Un ciclu de găurire are un **plan de poziționare** şi o **axă de găurire**. Planul de poziționare şi axa de găurire vor fi selectate cu instrucțiunile de selectare a planului G17, G18, G19.

Cod G	Plan de poziționare	Axa de găurire
G17	Planul X_p Y_p	Z_p
G18	Planul Z_p X_p	Y_p
G19	Planul Y_p Z_p	X_p

unde: X_p este axa X sau una paralelă cu ea.
Y_p este axa Y sau una paralelă cu ea.
Z_p este axa Z sau una paralelă cu ea.
Axele U, V, W sunt considerate axe paralele când sunt definite în parametri .

Ciclurile de găurire pot fi configurate cu instrucțiunile G98 şi G99.

G98 : Scula este retrasă până în punctul inițial în cursul ciclului de găurire. Este starea normală (implicită) în care trece comanda numerică după punerea sub tensiune, reset sau ştergerea modului ciclu.

G99 : Scula este retrasă până în punctul R în cursul ciclului de găurire; în consecință operațiile 9 şi 10 sunt omise.

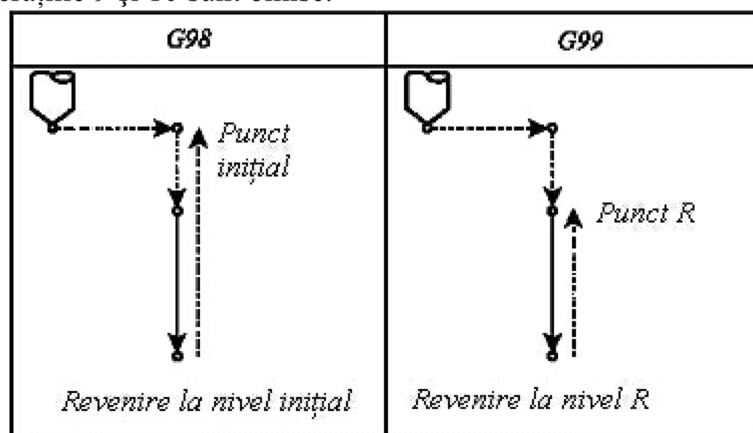


Fig. 17-2

Codurile ciclurilor de găurire: G73, G74, G76, G81, …,G89
Acestea vor impune modul ciclului particular validând variabilele ciclului pentru a fi modale. Codul G80 va anula modul ciclu şi va şterge variabilele ciclului din memorie.

Adresele folosite în **ciclurile de găurire** (şi semnificațiile lor) sunt:

G17	G_	Xp_	Yp_	I_	J_	Zp_	R_	Q_	E_	P_	F_	S_	L_
G18	G_	Zp_	Xp_	K_	I_	Yp_	R_	Q_	E_	P_	F_	S_	L_
G19	G_	Yp_	Zp_	J_	K_	Xp_	R_	Q_	E_	P_	F_	S_	L_
	(1)	(2)		(3)		(4)							(5)

(1) Codul de găurire
(2) Poziția găurii
(3) Deplasamentul după orientarea arborelui
(4) Datele de găurire
(5) Numărul de repetări

Codul găuririi:

În continuare sunt explicate semnificațiile codurilor.
Fiecare cod va fi modal până când este programată o instrucțiune G80 sau un cod care aparține grupului 1 de coduri (coduri de interpolare: G01, G02, G03, G33).
Atât timp cât starea de ciclu este activă (instrucțiunile G73, G74, G76,G81, …,G89), variabilele modale ale ciclului vor fi modale de asemenea şi între diversele tipuri de cicluri

de găurire.

Punct inițial:

Punctul inițial este poziția axei selectate pentru găurire; el va fi înregistrat
- Când este impus modul ciclu. De exemplu, în cazul

```
N1 G17 G90 G0 Z200
N2 G81 X0 Y0 Z50 R150
N3 X100 Y30 Z80
```

Poziția punctului inițial va fi Z=200 în blocurile N2 şi N3.
- Sau când este selectată o nouă axă de găurire. De exemplu:

```
N1 G17 G90 G0 Z200 W50
N2 G81 X0 Y0 Z50 R150
N3 X100 Y30 W20 R25
```

Poziția punctului inițial va fi Z=200 în blocul N2
Poziția punctului inițial va fi W=50 în blocul N3

Programarea lui R este obligatorie atunci când se schimbă selecția axei de găurire, altfel este afişat mesajul de eroare *3055 NO BOTTOM OR R POINT (nu s-a dat punct de fund sau R)*.

Poziția găurii - Xp, Yp, Zp

Dintre valorile coordonatelor introduse, vor fi considerate pentru poziția găurii cele din planul selectat.
Valorile introduse pot fi incrementale sau absolute, rectangulare (carteziene) sau polare, în unități metrice sau în inci.
Comenzile pentru imagine în oglindă, rotirea sistemului de coordonate şi modificare la scară sunt aplicabile valorilor coordonatelor introduse.
Comanda numerică deplasează scula la poziția găurii cu avans rapid indiferent de care cod din grupa 1 este activ.

Deplasarea după orientarea arborelui – I, J, K

Dacă maşina respectivă este prevăzută cu facilitatea de orientare a arborelui, scula poate fi ridicată de pe suprafața prelucrată la retragere în ciclurile de alezare fină şi alezare la retragere G76 şi G87 pentru a nu zgâria suprafața găurii. Direcția în care trebuie retrasă scula de pe suprafață poate fi specificată la adresele I, J, sau K. Comanda numerică va interpreta adresele în conformitate cu planul selectat.

G17: I, J
G18: K, I
G19: J, K

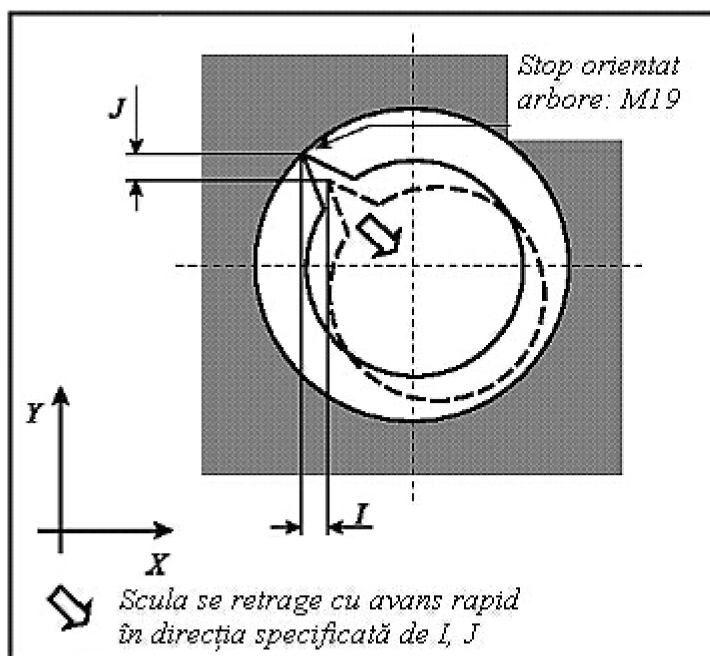


Fig. 17-3

Fiecare adresă este interpretată ca o dată incrementală cu coordonate rectangulare. Adresa poate fi în sistem metric sau în inci.
Comenzile de imagine în oglindă, rotire sistem de coordonate şi modificare la scară nu sunt aplicabile datelor I, J, K. Acestea din urmă sunt valori modale. Ele sunt şterse cu G80 sau cu codurile din grupul de interpolare. Retragerea este realizată cu avans rapid.

Datele de găurire

Poziția fundului găurii (punctul Z): X_p, Y_p, Z_p

Poziția fundului găurii sau a punctului Z (în cazul G17) trebuie specificată la adresa axei de găurire. Coordonata punctului de fund al găurii va fi întotdeauna interpretată ca dată rectangulară care poate fi specificată în unități metrice sau în inci, cu valoare absolută sau incrementală. Când valoarea punctului de fund este specificată incremental, deplasarea va fi calculată din punctul R.

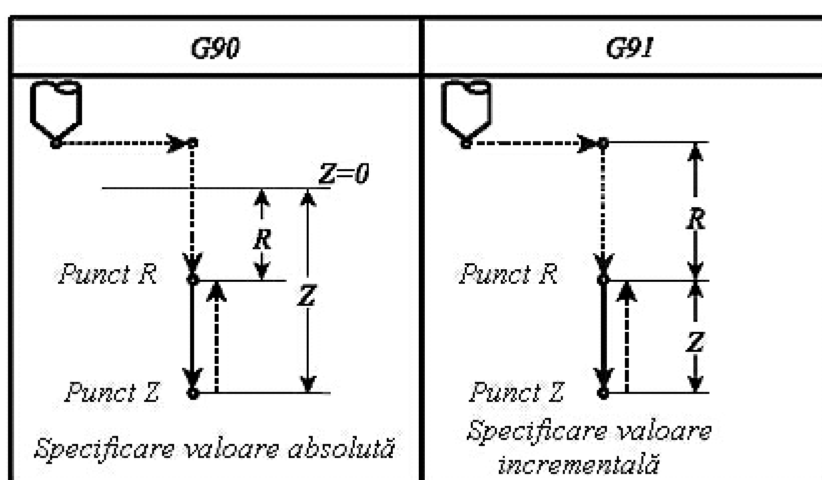


Fig. 17-4

Comenzile de imagine în oglindă şi modificare la scară sunt aplicabile datelor punctului de fund. Acestea din urmă sunt valori modale. Ele sunt şterse cu G80 sau cu codurile din grupul de interpolare. Comanda numerică va face apropierea de punctul Z cu valoarea avansului care este activă.

Punctul R

Punctul de apropiere este specificat la adresa R. Este întotdeauna o dată de coordonată rectangulară care poate fi specificată în unități metrice sau în inci, cu valoare absolută sau incrementală. Comenzile de imagine în oglindă şi modificare la scară sunt aplicabile datelor punctului R. Acestea din urmă sunt valori modale. Ele sunt şterse cu G80 sau cu codurile din grupul de interpolare. Comanda numerică va face apropierea de punctul R cu avans rapid.

Valoarea de intrare a sculei (Q)

Este adâncimea de intrare, în ciclurile G73 şi G83. Este invariabil o dată incrementală, pozitivă rectangulară (modală). Valoarea ei va fi ştearsă cu G80 sau cu codurile din grupul de interpolare. Comanda de modificare la scară nu afectează valoarea adâncimii de intrare.

Data auxiliară (E)
Mărimea retragerii în ciclul G73 şi valoarea de curăţare în ciclul G83 este specificată la adresa E. Este întotdeauna o dată incrementală, rectangulară, pozitivă. Comanda de modificare la scară nu are efect asupra datei auxiliare. Este valoare modală. Valoarea ei va fi ştearsă cu G80 sau cu codurile din grupul de interpolare. Dacă nu a fost programată, comanda numerică preia valoarea necesară din parametrul *RETG73*, sau *CLEG83*.

Întârzierea (P)
Specifică timpul de întârziere la fundul găurii. Specificarea sa se face după regulile descrise la G04. Este valoare modală. Valoarea ei va fi ştearsă cu G80 sau cu codurile din grupul de interpolare.

Avansul (F)
Va defini avansul. Este valoare modală, rescrisă doar de programarea unei alte date F. Nu va fi ştearsă de G80 sau alt cod.

Viteza arborelui (S)
Valoare modală, rescrisă doar de programarea unei alte date S. Nu va fi ştearsă de G80 sau alt cod.

Numărul de repetări (L)
Defineşte numărul de repetări ale ciclului în gama de la 1 la 9999. Dacă L nu este completat, este presupus L=1. În cazul L=0 datele ciclului vor fi stocate dar nu se vor executa. Valoarea pentru L este activă doar în blocul în care a fost specificată.

Exemple de coduri de găurire modale şi de variabile de ciclu:

```
N1 G17 G0 Z_ M3
N2 G81 X_ Y_ Z_ R_ F_
```
Este obligatoriu să se specifice datele de găurire (Z, R) la începutul modului ciclu.
```
N3 X__
```
Întrucât datele de găurire au fost specificate în blocul N2 şi sunt folosite neschimbate în N3, nu mai trebuie completate din nou, adică se poate omite G81, Z_, R_, F_. Poziţia găurii este variată doar în direcţia X, scula se mişcă în acea direcţie şi va găuri aceeaşi gaură ca în blocul N2.
```
N4 G82 Y_ Z_  P_
```
Poziţia găurii se schimbă în direcţia Y. Metoda de găurire se conformează cu G82, punctul de fund ia o valoare nouă (Z), punctul R şi avansul (R, F) sunt luate din blocul N2.
```
N2 G80 M5
```
Modul ciclu şi variabilele modale ale ciclului (cu excepţia F) vor fi şterse.
```
N6 G85 Y_ Z_ R_ P_ M3
```
Deoarece datele de găurire au fost şterse în blocul N5 cu comanda G80, trebuie specificate din nou valorile Z, R şi P.
```
N7 G0 X_ Y_
```
Modul ciclu şi variabilele modale ale ciclului (cu excepţia F) vor fi şterse.

Exemplu de utilizare a repetării ciclului:
Dacă trebuie găurite un anumit tip de găuri cu parametri neschimbați în poziții egal distanțate, se poate specifica numărul de repetări la adresa L. Valoarea lui L este activă doar în acel bloc în care a fost specificată.

```
N1 G90 G19 G0 X0 Y0 Z100 M3
N2 G91 G81 X100 Z-40 R-97 F50 L5
```

Cu instrucțiunile de mai sus se vor face 5 găuri identice distanțate la 100mm pe axa X. Poziția primei găuri este X=100, Y=0. Pentru G91 poziția găurii trebuie specificată incremental. Dacă s-a specificat ca dată absolută (G90), operația ar fi fost executată de cinci ori succesiv în punctul de coordonate X100, Y0.

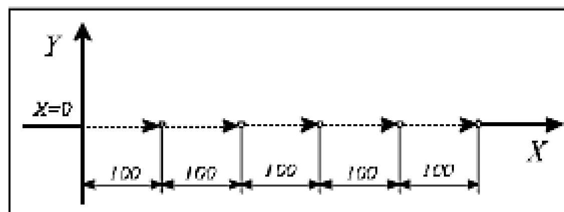


Fig. 17-5

```
N1 G90 G17 G16 G0 X200 Y-60 Z50 M3
N2 G81 YI60 Z-40 R3 F50 L6
```

Cu instrucțiunile de mai sus se vor face 6 găuri distanțate la 60 de grade pe un cerc cu raza de 200mm. Poziția primei găuri coincide cu punctul de coordonate X=200 Y=0.

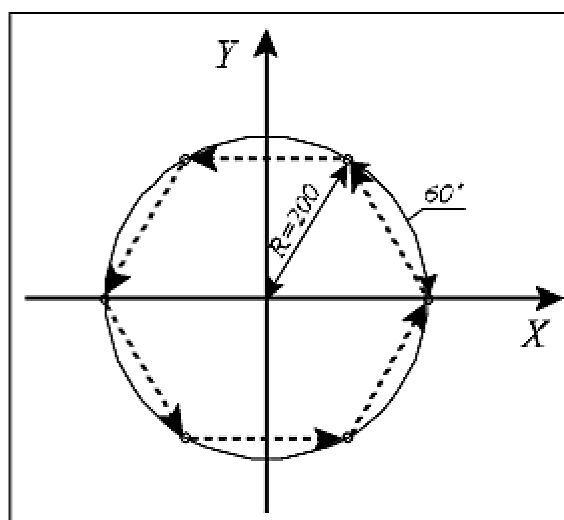


Fig. 17-6

17.1 Descrierea detaliată a ciclurilor memorate

17.1.1 Ciclu de găurire cu viteză mare şi retrageri pentru ruperea şpanului (G73)

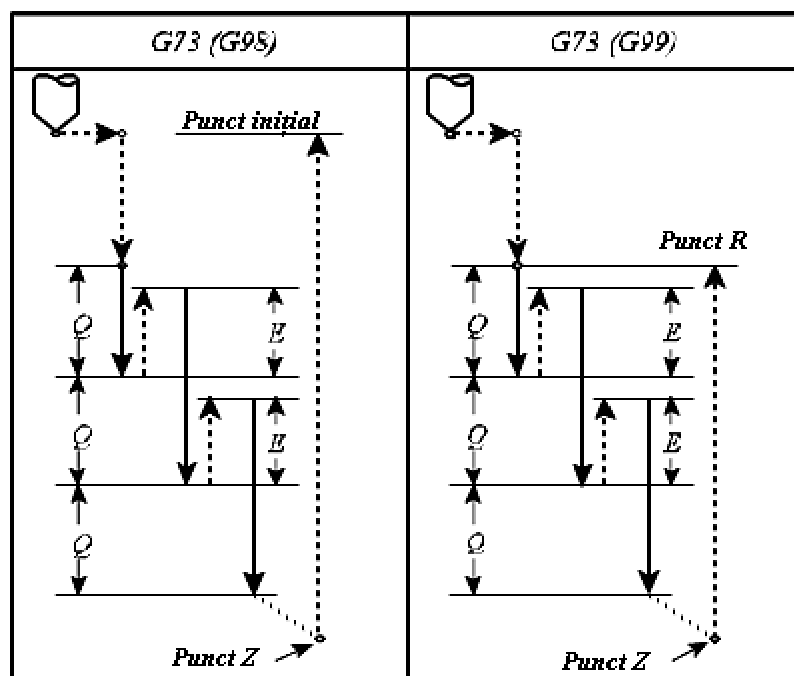


Fig. 17.1.1-1

Variabilele folosite în ciclu sunt:

G17 **G73** X_p__ Y_p__ Z_p__ R__ Q__ E__ F__ L__
G18 **G73** Z_p__ X_p__ Y_p__ R__ Q__ E__ F__ L__
G19 **G73** Y_p__ Z_p__ X_p__ R__ Q__ E__ F__ L__

Operațiile ciclului sunt:

1. poziționare cu avans rapid
2. -
3. deplasare cu avans rapid în punctul R
4. -
5. găurire până în punctul Z cu avansul F
6. -
7. cu G99, retragere în punctul R, cu avans rapid
8. -
9. cu G98, retragere în punctul inițial, cu avans rapid
10. -

Descrierea operației de găurire 5 este după cum urmează:
- găurirea în piesă cu avans de lucru a adâncimii de intrare specificate la adresa Q,
- retragerea cu avans rapid pe distanța specificată la adresa F sau în parametrul *RETG73*,
- găurirea din nou a adâncimii de intrare Q, calculată din punctul final al intrării anterioare,
- retragerea cu avans rapid la valoarea specificată la adresa E.

Procedura descrisă mai sus este executată până la punctul de fund specificat la adresa Z.

17.1.2 Ciclu de filetare în sens antiorar (G74)

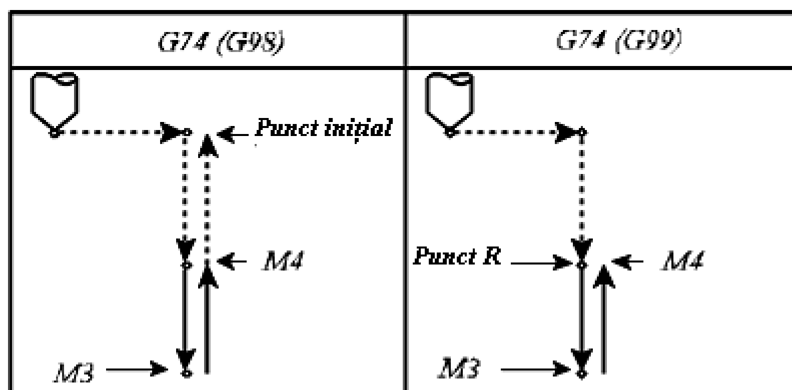


Fig. 17.1.2-1

Acest ciclu poate fi folosit doar cu un tarod cu resort. Variabilele utilizate sunt:

G17 **G74** X_p__ Y_p__ Z_p__ R__ (P__) F__ L__
G18 **G74** Z_p__ X_p__ Y_p__ R__ (P__) F__ L__
G19 **G74** Y_p__ Z_p__ X_p__ R__ (P__) F__ L__

Înainte de a se porni ciclul, arborele trebuie pornit sau programat să se rotească în direcția lui M4 (sens antiorar).
Valoarea avansului trebuie specificată în conformitate cu pasul filetului tarodului.
- În starea G94 (avans pe minut):

$$F=P*S$$

unde P este pasul filetului în mm/rot sau inci/rot,
S este viteza arborelui în rot/min.
- În starea G95 (avans pe rotație):

$$F=P$$

unde P este pasul filetului în mm/rot sau inci/rot,
Operațiile ciclului sunt:
1. poziționare cu avans rapid în planul selectat
2. -
3. deplasare cu avans rapid în punctul R
4. -
5. găurire până în punctul de fund, cu avansul F (corecția şi stopul sunt inactive)
6. întârziere cu valoarea specificată la adresa P, în cazul în care parametrul *TAPDWELL* este validat (=1), inversare sens arbore (M3)
7. retragere în punctul R, cu avansul F (corecția şi stopul sunt inactive)
8. inversare sens arbore (M4)
9. cu G98, retragere în punctul inițial, cu avans rapid
10. -

17.1.3 Ciclu de alezare fină (G76)

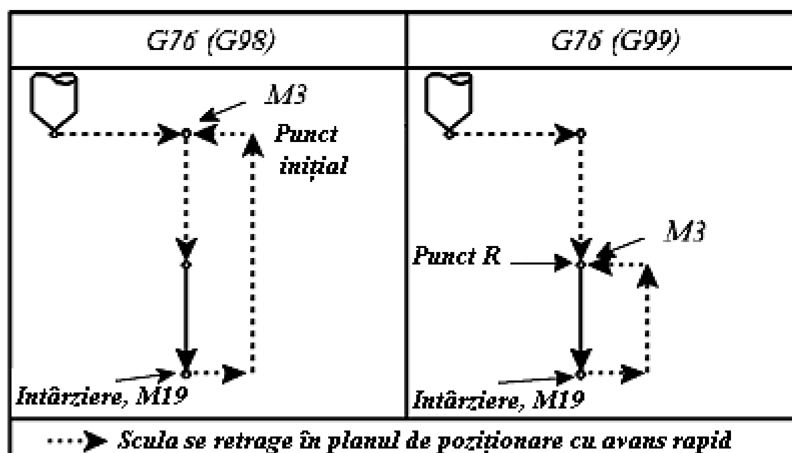


Fig. 17.1.3-1

Ciclul G76 este aplicabil doar când maşina unealtă este prevăzută cu facilitatea de orientare arbore. În aceste caz parametrul ORIENT1 trebuie să fie impus 1, altfel este afişat mesajul *ERROR IN G76*.
Întrucât în punctul de fund ciclul execută orientarea arborelui şi îndepărtează scula de pe suprafața prelucrată cu valorile specificate la I, J şi K, piesa nu va fi zgâriată atunci când scula se retrage.
Variabilele folosite în ciclu sunt:

G17 **G76** X_p__ Y_p__ I__ J__ Z_p__ R__ P__ F__ L__
G18 **G76** Z_p__ X_p__ K__ I__ Y_p__ R__ P__ F__ L__
G19 **G76** Y_p__ Z_p__ J__ K__ X_p__ R__ P__ F__ L__

Comanda M3 trebuie transmisă înainte de pornirea ciclului.
Operațiile ciclului sunt:

1. poziționare cu avans rapid în planul selectat
2. -
3. deplasare cu avans rapid în punctul R
4. -
5. găurire până în punctul Z, cu avansul F
6. - întârziere cu valoarea specificată la adresa P
   - orientare arbore (M19)
   - retragere sculă cu avans rapid cu valorile I, J, K în planul selectat
7. cu G99, retragere în punctul R, cu avans rapid
8. cu G99,
   - retragere sculă cu avans rapid în planul selectat, invers față de valorile specificate la I, J, K
   - arbore repornit în direcția M3
9. cu G98, retragere în punctul inițial, cu avans rapid
10. cu G98,
   - retragere sculă cu avans rapid în planul selectat, invers față de valorile specificate la I, J, K
   - arbore repornit în direcția M3

17.1.4 Ciclu memorat pentru anularea găuririi (G80)

Codul **G80** va anula starea de ciclu, variabilele ciclului fiind şterse.
Z şi R vor lua valorile incrementale 0 (restul variabilelor vor lua valoarea 0).
Dacă în blocul G80 se introduc coordonate dar fără să se introducă alte instrucțiuni, se execută deplasarea cu codurile de interpolare care erau active înainte de activarea ciclului (cod G grupa 1).

17.1.5 Găurire, ciclu de alezare parțială (G81)

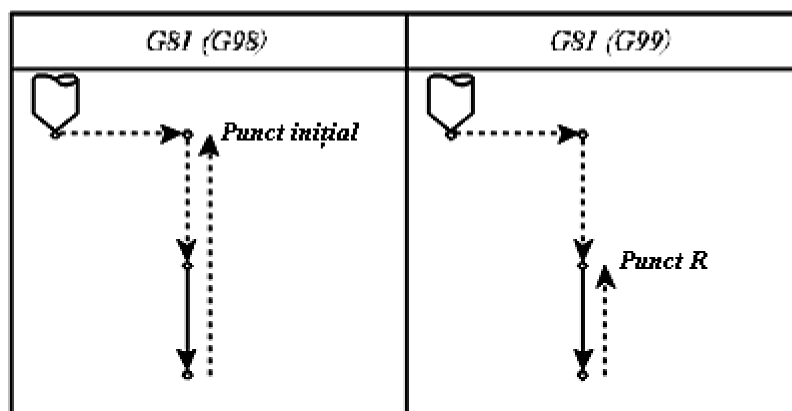


Fig. 17.1.5-1

Variabilele folosite în ciclu sunt:

G17 **G81** X_p__ Y_p__ Z_p__ R__ F__ L__
G18 **G81** Z_p__ X_p__ Y_p__ R__ F__ L__
G19 **G81** Y_p__ Z_p__ X_p__ R__ F__ L__

Operațiile ciclului sunt:

1. poziționare cu avans rapid în planul selectat
2. -
3. deplasare cu avans rapid în punctul R
4. -
5. găurire până în punctul Z, cu avansul F
6. -
7. cu G99, retragere în punctul R, cu avans rapid
8. -
9. cu G98, retragere în punctul inițial, cu avans rapid
10. -

17.1.6 Găurire, ciclu de alezare la retragere (G82)

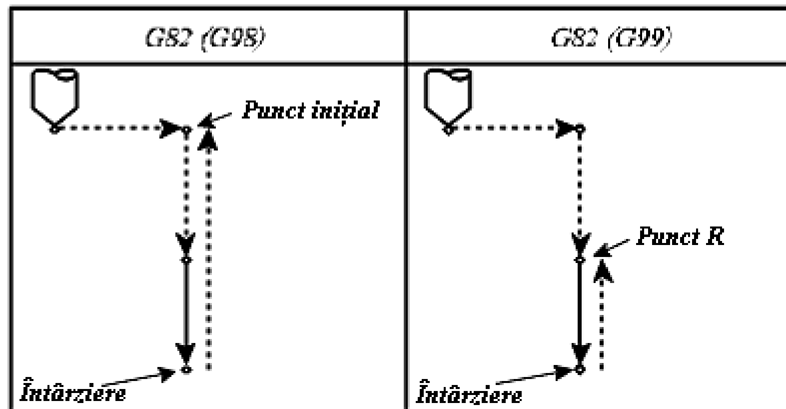


Fig. 17.1.6-1

Variabilele folosite în ciclu sunt:

G17 **G82** X_p__ Y_p__ Z_p__ R__ P__ F__ L__
G18 **G82** Z_p__ X_p__ Y_p__ R__ P__ F__ L__
G19 **G82** Y_p__ Z_p__ X_p__ R__ P__ F__ L__

Operațiile ciclului sunt:

1. poziționare cu avans rapid în planul selectat
2. -
3. deplasare cu avans rapid în punctul R
4. -
5. găurire până în punctul de fund, cu avansul F
6. întârziere cu timpul specificat la adresa P
7. cu G99, retragere în punctul R, cu avans rapid
8. -
9. cu G98, retragere în punctul inițial, cu avans rapid
10. -

17.1.7 Ciclu de găurire cu retrageri pentru ruperea şpanului (G83)

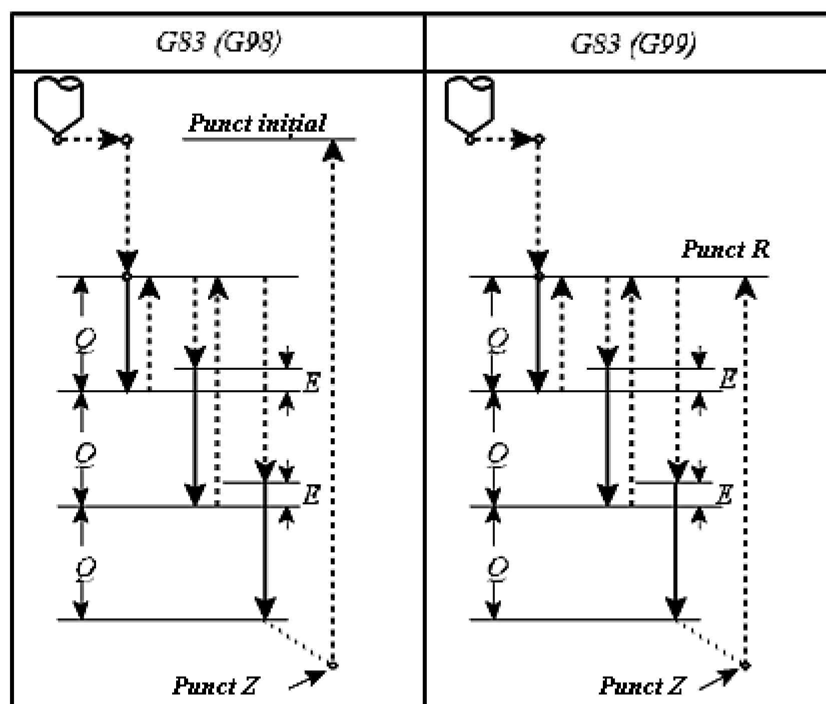


Fig. 17.1.7-1

Variabilele folosite în ciclu sunt:

G17 **G83** X_p__ Y_p__ Z_p__ R__ Q__ E__ F__ L__
G18 **G83** Z_p__ X_p__ Y_p__ R__ Q__ E__ F__ L__
G19 **G83** Y_p__ Z_p__ X_p__ R__ Q__ E__ F__ L__

Operațiile ciclului sunt:

1. poziționare cu avans rapid
2. -
3. deplasare cu avans rapid în punctul R
4. -
5. găurire până în punctul Z cu avansul F
6. -
7. cu G99, retragere în punctul R, cu avans rapid
8. -
9. cu G98, retragere în punctul inițial, cu avans rapid
10. -

Descrierea operației de găurire 5 este după cum urmează:

- Găurirea cu avans de lucru la adâncimea specificată la adresa Q,
- retragerea cu avans rapid în punctul R,
- apropierea cu avans rapid de adâncimea anterioară până la distanța de siguranță specificată la adresa E
- găurirea din nou la adâncimea Q, calculată din punctul final al intrării anterioare, cu avansul F (deplasament E+Q)
- retragerea cu avans rapid în punctul R.

Procedura descrisă mai sus este executată până la punctul de fund specificat la adresa Z.
Distanța E va fi luată din program (adresa E) sau de la parametrul *CLEG83*.

17.1.8 Ciclu de filetare cu tarod (G84)

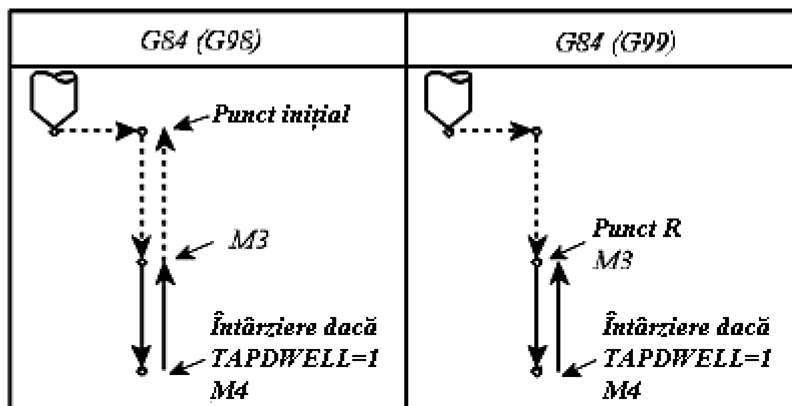


Fig. 17.1.8-1

Acest ciclu poate fi folosit doar cu un tarod cu resort.
Variabilele utilizate sunt:

G17 **G84** X_p__ Y_p__ Z_p__ R__ (P__) F__ L__
G18 **G84** Z_p__ X_p__ Y_p__ R__ (P__) F__ L__
G19 **G84** Y_p__ Z_p__ X_p__ R__ (P__) F__ L__

Înainte de a se porni ciclul, trebuie selectat sensul M3 (orar) de rotire a arborelui.
Valoarea avansului trebuie specificată în conformitate cu pasul filetului tarodului.

- În starea G94 (avans pe minut):

$$F=P*S$$

unde P este pasul filetului în mm/rot sau inci/rot,
S este viteza arborelui în rot/min.

- În starea G95 (avans pe rotație):

$$F=P$$

unde P este pasul filetului în mm/rot sau inci/rot,

Operațiile ciclului sunt:

1. poziționare cu avans rapid în planul selectat
2. -
3. deplasare cu avans rapid în punctul R
4. -
5. filetare până în punctul de fund, cu avansul F (corecția şi stopul sunt inactive)
6. întârziere cu valoarea specificată la adresa P, în cazul în care parametrul *TAPDWELL* este validat (=1), inversare sens arbore (M3)
7. retragere în punctul R, cu avansul F (corecția şi stopul sunt inactive)
8. inversare sens arbore (M4)
9. cu G98, retragere în punctul inițial, cu avans rapid
10. -

17.1.9 Cicluri de filetare rigidă cu tarod (în sens orar şi antiorar) (G84.2, G84.3)

Într-un ciclu de filetare raportul dintre avansul axei de găurire şi numărul de rotații pe minut al arborelui trebuie să fie egal cu pasul filetului tarodului. Cu alte cuvinte, în condiții ideale de filetare, raportul

$P = F/S$ trebuie să fie constant în orice moment,

unde: P este pasul filetului (mm/rot sau inci/rot),
F este avansul (mm/min sau inci/min),
S este viteza arborelui (rot/min).

Viteza de rotație a arborelui şi avansul axei de filetare sunt controlate în mod complet independent în ciclurile de filetare pe stânga G74, respectiv pe dreapta G84. Prin urmare, condiția de mai sus nu poate fi îndeplinită cu maximă acuratețe. Aceasta este valabil mai ales în fundul găurii unde avansul axei de găurire şi viteza de rotație a arborelui trebuie să fie reduse şi oprite în sincronism, iar apoi accelerate la fel în direcție opusă.
Problema de mai sus poate fi eliminată cu un tarod cu resort, aceasta compensând fluctuațiile în valoarea coeficientului *F/S*.
Un principiu diferit de control este adoptat în ciclurile de găurire G84.2 şi G84.3 permițând un tarod rigid (filetare fără resort). In acest caz, comanda numerică menține coeficientul *F/S* constant în orice moment.
În primul caz comanda numerică reglează doar viteza arborelui, în al doilea caz se controlează şi poziția sa. Mişcările axei de găurire şi ale arborelui sunt legate prin interpolare liniară în ciclurile G84.2 şi G84.3. În acest fel coeficientul *F/S* poate fi menținut constant şi în etapele de accelerare şi frânare.

G84.2: ciclu de filetare rigid cu tarod
G84.3: ciclu de filetare rigid cu tarod în sens invers

Ciclurile de mai sus sunt aplicabile doar pentru maşini la care arborele este prevăzut cu codor de poziție şi acționarea principală are posibilitatea de control al poziției (parametrul *INDEX1*=1). Dacă nu este îndeplinită condiția, se afişează mesajul de eroare *3052 ERROR IN G76, G87* atunci când se apelează acest mod.

Variabilele folosite în ciclu sunt

G17 **G84.** X_p__ Y_p__ Z_p__ R__ F__ S__ L__
G18 **G84.** Z_p__ X_p__ Y_p__ R__ F__ S__ L__
G19 **G84.** Y_p__ Z_p__ X_p__ R__ F__ S__ L__

Arborele se opreşte la sfârşitul ciclului; dacă e necesar, trebuie să fie repornit de către programator.
Avansul şi viteza de rotire a arborelui trebuie să fie specificate în conformitate cu pasul filetului tarodului.

- În starea G94 (avans pe minut), *F=P*S*
  unde P este pasul filetului în mm/rot sau inci/rot,
  S este viteza arborelui în rot/min.
  În acest caz deplasamentul şi avansul în lungul axei de găurire şi arborele vor fi după cum urmează (Z presupusă a fi axa de găurire)

	deplasament	***avans***
Z	*z= distanța dintre punctele R și Z*	$F_z = F \left(\frac{mm}{min} \text{ sau } \frac{inci}{min} \right)$
S	$s = \frac{z \cdot S \cdot 360}{F}$ (grade)	$F_s = S \cdot 360 \left(\frac{(grade)}{min} \right)$

- În starea G95 (avans pe rotație), *F = P*
  unde P este pasul filetului în mm/rot sau inci/rot. Evident că în starea G95 (avans pe rotație) se poate programa direct pasul filetului, dar pentru a defini avansul trebuie programat şi S.
  În acest caz, deplasamentul şi avansul în lungul axei de găurire şi arborele vor fi după cum urmează (Z presupusă a fi axa de găurire):

	deplasament	***avans***
Z	*z=R distanța dintre punct și punctul de bază*	$F_z = F \cdot S \left(\frac{mm}{min} \text{ sau } \frac{inci}{min} \right)$
S	$s = \frac{z \cdot 360}{F}$ (grade)	$F_s = S \cdot 360 \left(\frac{grade}{min} \right)$

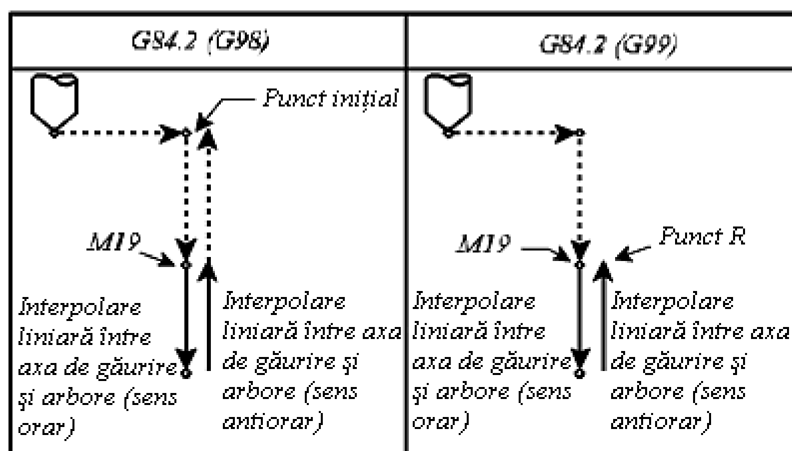


Fig. 17.1.9-1

În cazul G84.2, operațiile ciclului sunt:
1. poziționare cu avans rapid în planul selectat
2. -
3. deplasare cu avans rapid în punctul R
4. orientarea arborelui (M19)

5. interpolare liniară între axa de găurire şi arbore, cu arborele rotindu-se în sens orar
6. -
7. interpolare liniară între axa de găurire şi arbore, cu arborele rotindu-se în sens antiorar
8. -
9. cu G98, retragere cu avans rapid în punctul inițial
10. -

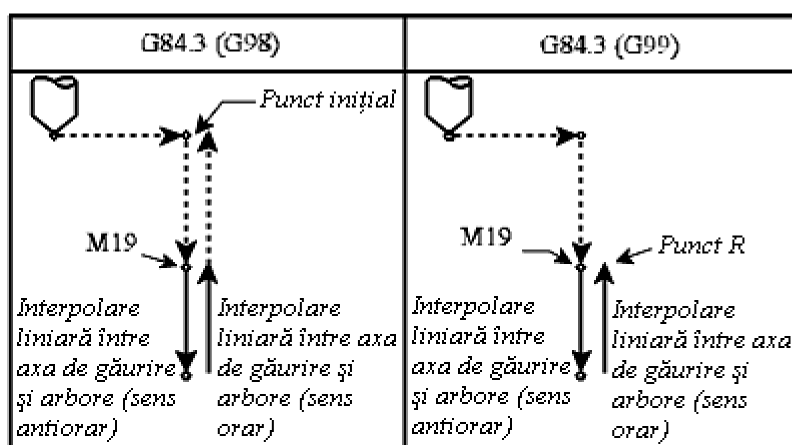


Fig. 17.1.9-2

În cazul G84.3, operațiile ciclului sunt:

1. poziționare cu avans rapid în planul selectat
2. -
3. deplasare cu avans rapid în punctul R
4. orientarea arborelui (M19)
5. interpolare liniară între axa de găurire şi arbore, cu arborele rotindu-se în sens antiorar (-)
6. -
7. interpolare liniară între axa de găurire şi arbore, cu arborele rotindu-se în sens orar (+)
8. -
9. cu G98, retragere cu avans rapid în punctul inițial
10. -

17.1.10 Ciclu de alezare (G85)

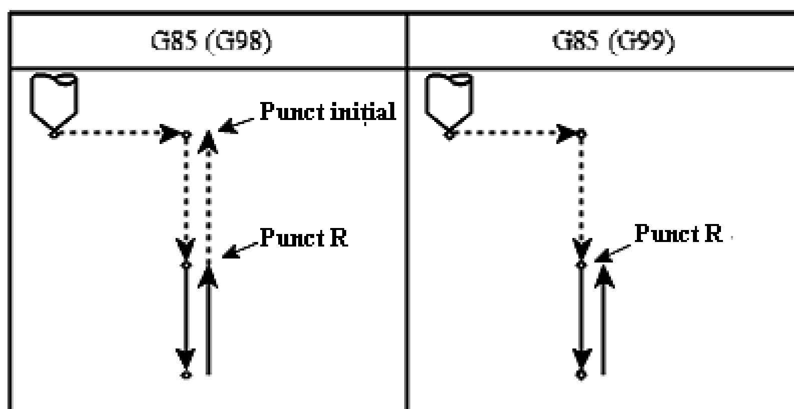


Fig. 17.1.10-1

Variabilele folosite în ciclu sunt:

G17 **G85** X_p__ Y_p__ Z_p__ R__ F__ L__
G18 **G85** Z_p__ X_p__ Y_p__ R__ F__ L__
G19 **G85** Y_p__ Z_p__ X_p__ R__ F__ L__

Operațiile ciclului sunt:

1. poziționare cu avans rapid în planul selectat
2. -
3. deplasare cu avans rapid în punctul R
4. -
5. alezare până în punctul de fund, cu avansul F
6. -
7. retragere în punctul R, cu avansul F
8. -
9. cu G98, retragere cu avans rapid în punctul inițial
10. -

17.1.11 Ciclu de alezare cu retragerea sculei cu avans rapid (G86)

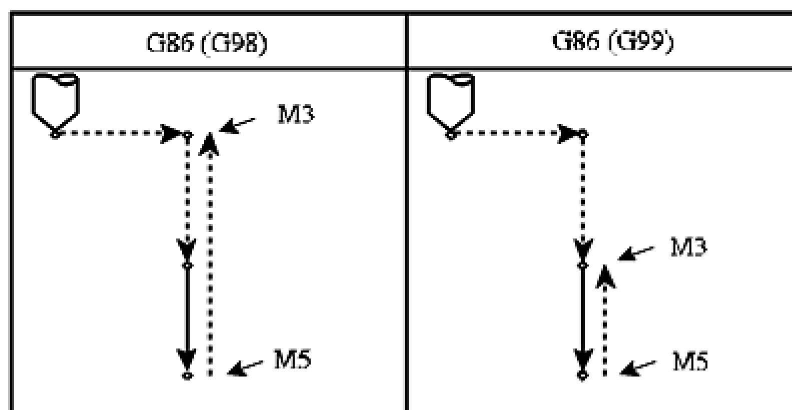


Fig. 17.1.11-1

Variabilele folosite în ciclu sunt:

G17 **G86** X_p__ Y_p__ Z_p__ R__ F__ L__
G18 **G86** Z_p__ X_p__ Y_p__ R__ F__ L__
G19 **G86** Y_p__ Z_p__ X_p__ R__ F__ L__

Comanda M3 trebuie introdusă înainte de pornirea ciclului.

Operațiile ciclului sunt:

1. poziționare cu avans rapid în planul selectat
2. -
3. deplasare cu avans rapid în punctul R
4. -
5. alezare până în punctul Z cu avansul F
6. oprire arbore (M5)
7. cu G99, retragere cu avans rapid în punctul R
8. cu G99, repornire arbore (M3)
9. cu G98, retragere cu avans rapid în punctul inițial
10. cu G98, repornire arbore (M3)

17.1.12 Ciclu de alezare / ciclu de alezare la retragere (G87)

Ciclul va fi executat în două moduri diferite.

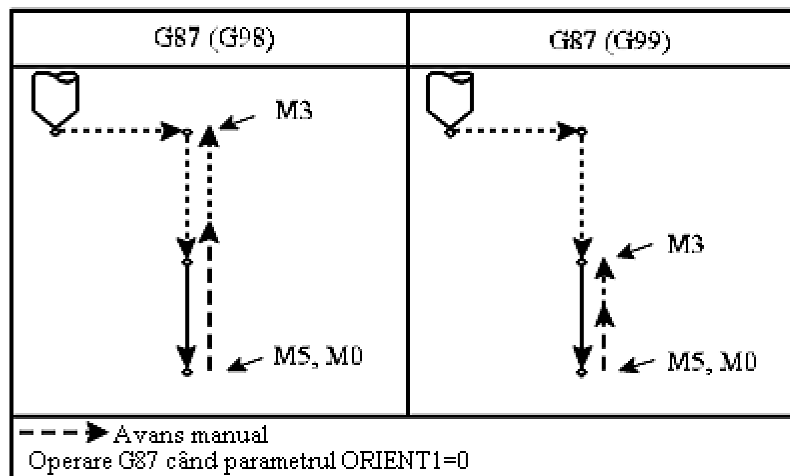


Fig. 17.1.12-1

A. Ciclu de alezare, operare manuală în punctul de fund
Dacă maşina nu este prevăzută cu facilitatea de orientare arbore (parametrul *ORIENT1*=0), comanda numerică va merge pe varianta „A“.
Variabilele folosite în ciclu sunt:

G17 **G87** X_p__ Y_p__ Z_p__ R__ F__ L__
G18 **G87** Z_p__ X_p__ Y_p__ R__ F__ L__
G19 **G87** Y_p__ Z_p__ X_p__ R__ F__ L__

Comanda M3 trebuie introdusă înainte de pornirea ciclului.
Operațiile ciclului sunt:

1. poziționare cu avans rapid în planul selectat
2. -
3. deplasare cu avans rapid în punctul R
4. -
5. alezare până în punctul Z cu avansul F
6. - oprire arbore (M5)
   - comanda numerică trece în starea STOP (M0), din care operatorul poate ajunge într-unul din modurile de deplasare manuală (JOG, INCREMENTAL JOG , sau HANDLE (roată de mână)) şi să execute operarea manuală, de exemplu să retragă scula de la peretele găurii şi apoi să o extragă din gaură. După revenirea în modul AUTO se poate continua prelucrarea cu START.
7. cu G99, START urmat de retragere cu avans rapid în punctul R
8. cu G99, repornire arbore (M3)
9. cu G98, START urmat de retragere cu avans rapid în punctul inițial
10. cu G98, repornire arbore (M3)

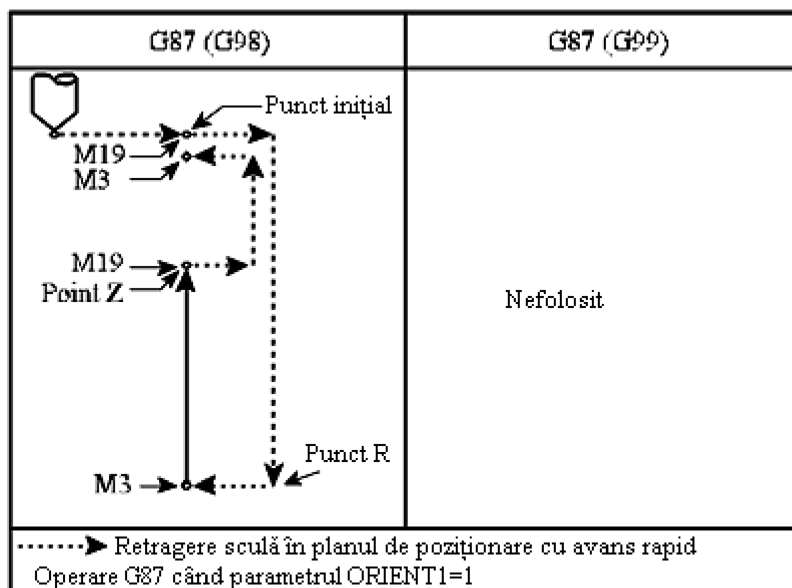


Fig. 17.1.12-2

B. Ciclu de alezare la retragere
Dacă maşina este prevăzută cu facilitatea de orientare arbore (parametrul *ORIENT1*=1), comanda numerică va merge pe varianta „B".
Variabilele folosite în ciclu sunt:

G17 **G87** X_p__ Y_p__ I__ J__ Z_p__ R__ F__ L__
G18 **G87** Z_p__ X_p__ K__ I__ Y_p__ R__ F__ L__
G19 **G87** Y_p__ Z_p__ J__ K__ X_p__ R__ F__ L__

Comanda M3 trebuie introdusă înainte de pornirea ciclului.
Operațiile ciclului sunt:

1. poziționare cu avans rapid în planul selectat
2. - orientare arbore
   - retragere sculă în planul selectat cu valorile I, J, K (avans rapid)
3. deplasare cu avans rapid în punctul R
4. - retragere sculă în planul selectat, invers față de valorile specificate la I, J sau K (avans rapid)
   - arbore repornit în direcția M3
5. alezare până în punctul Z, cu avansul F
6. - orientare arbore (M19)
   - retragere sculă în planul selectat cu valorile I, J, K (avans rapid)
7. -
8. -
9. retragere cu avans rapid în punctul inițial
10. - retragere sculă în planul selectat, invers față de valorile specificate la I, J sau K (avans rapid)
    - arbore repornit în direcția M3

Spre deosebire de cazurile anterioare, punctul R este localizat mai jos decât punctul Z. Aceasta trebuie avut în vedere la programarea axei de alezare şi a adresei R.

17.1.13 Ciclu de alezare (operare manuală în punctul de fund) (G88)

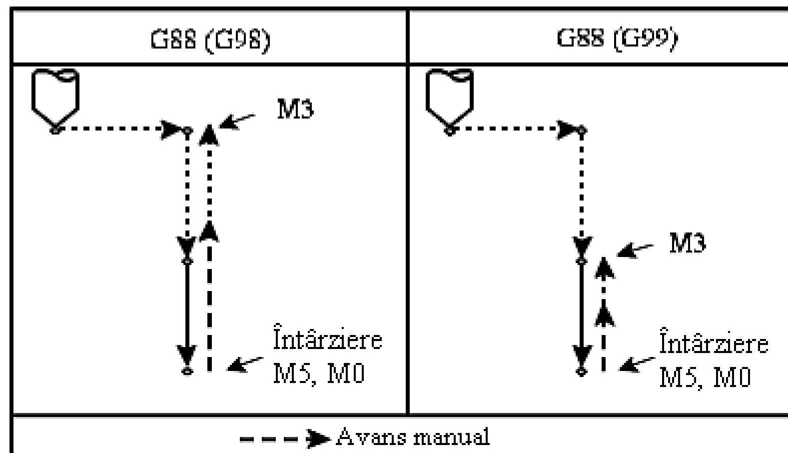


Fig. 17.1.13-1

Variabilele folosite în ciclu sunt:

G17 **G88** X_p__ Y_p__ Z_p__ R__ P__ F__ L__
G18 **G88** Z_p__ X_p__ Y_p__ R__ P__ F__ L__
G19 **G88** Y_p__ Z_p__ X_p__ R__ P__ F__ L__

Comanda M3 trebuie introdusă înainte de pornirea ciclului.
Operațiile ciclului sunt:

1. poziționare cu avans rapid în planul selectat
2. -
3. deplasare cu avans rapid în punctul R
4. -
5. alezare până în punctul de fund cu avansul F
6. - întârziere cu valoarea P
   - comanda numerică trece în starea STOP (M0), din care operatorul poate ajunge într-unul din modurile de deplasare manuală (JOG, INCREMENTAL JOG , sau HANDLE (roată de mână)) şi să execute prelucrarea manual, de exemplu să retragă scula de la peretele găurii şi apoi să o extragă din gaură. După întoarcerea la modul AUTO se poate continua prelucrarea cu START.
7. cu G99, START urmat de retragere cu avans rapid în punctul R
8. cu G99, repornire arbore (M3)
9. cu G98, START urmat de retragere cu avans rapid în punctul inițial
10. cu G98, repornire arbore (M3)

Ciclul este acelaşi ca în cazul „A“ al G87 dar cu întârziere înainte de oprirea arborelui.

17.1.14 Ciclu de alezare (întârziere în punctul de fund, retragere cu avans de lucru) (G89)

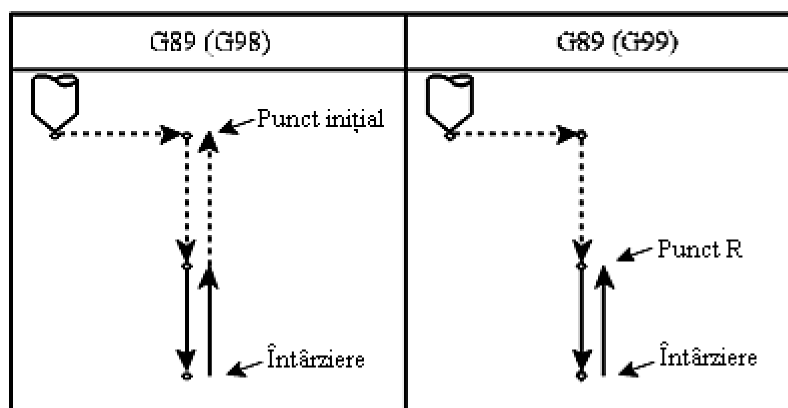


Fig. 17.1.14-1

Variabilele folosite în ciclu sunt:

G17 **G89** X_p__ Y_p__ Z_p__ R__ P__ F__ L__
G18 **G89** Z_p__ X_p__ Y_p__ R__ P__ F__ L__
G19 **G89** Y_p__ Z_p__ X_p__ R__ P__ F__ L__

Operațiile ciclului sunt:

1. poziționare cu avans rapid în planul selectat
2. -
3. deplasare cu avans rapid în punctul R
4. -
5. alezare până în punctul de fund, cu avansul F
6. întârziere cu valoarea specificată la adresa P
7. retragere în punctul R, cu avansul F
8. -
9. cu G98, retragere cu avans rapid în punctul inițial
10. -

Cu excepția întârzierii, ciclul este identic cu G85.

17.2 Note cu privire la folosirea ciclurilor memorate pentru găurire

- Ciclul de găurire va fi executat în modul ciclu prevăzând un bloc fără cod G care să conțină una din adresele

  X_p, Y_p, Z_p sau R

  Altfel ciclul de găurire nu va fi executat.
- Având programată întârzierea G04 P în modul ciclu, comanda va fi executată în conformitate cu P programat, dar variabila de întârziere a ciclului **nu** va fi ştearsă şi **nu** va fi reînscrisă.
- Valorile pentru I, J, K, Q, E, P trebuie să fie specificate în blocul în care se execută găurirea, altfel valorile **nu** vor fi stocate.

Pentru a ilustra cele spuse mai sus, să examinăm următorul exemplu:

```
G81 X_ Y_ Z_ R_ F     (ciclul de găurire este executat)
    X                 (ciclul de găurire este executat)
    F_                (ciclul de găurire nu este executat,
                       F este supraînscris)
    M_                (ciclul de găurire nu este executat,
                       codul M este executat)
G4  P_                (ciclul de găurire nu este executat,
                       întârzierea va fi rescrisă, dar nu
                 valoarea întârzierii variabilei ciclului)
    I_ Q_             (ciclul de găurire nu este executat,
                       valorile programate nu vor fi
                 înregistrate ca variabile ale ciclului)
```

- Dacă se programează într-un bloc o funcție precum şi un ciclu de găurire, funcția va fi executată la sfârşitul primei operații, după terminarea poziționării. Dacă a fost programat şi L în ciclu, funcția va fi executată numai la prima execuție.

- În modul bloc cu bloc, comanda numerică se va opri în timpul ciclului după fiecare din operațiile 1, 3 şi 10.

- Butonul de STOP nu este eficace la fiecare din operațiile 5, 6 şi 7 din ciclurile G74, G84. Dacă este acționat STOP pe durata acestor operații, se va continua prelucrarea până la sfârşitul operației 7.

- Corecțiile pentru avans şi arbore vor fi întotdeauna 100% în fiecare din operațiile 5, 6 şi 7 ale ciclurilor G74, G84 indiferent de pozițiile comutatoarelor de corecție.

- Dacă este programat G43, G44, G49 într-un ciclu de interpolare, sau dacă este specificată o nouă valoare pentru H, lungimea de compensare este luată în considerație în operația 3, în mod invariabil de-a lungul axei de găurire.

- Instrucțiunile G45, … G48 nu vor fi executate în ciclul de găurire.

18 Funcții de măsurare

18.1 Funcția de salt (G31)

Instrucțiunea
G31 v (F) (P)
porneşte interpolarea liniară către punctul cu coordonata v. Mişcarea se execută până când se primeşte un semnal extern de salt (de exemplu cel de la un sesizor de atingere) sau se ajunge la poziția finală specificată la coordonata lui v. Comanda numerică va încetini şi apoi va opri mişcarea după sosirea semnalului de salt.
Adresa P specifică o intrare de semnal de salt dintre cele patru disponibile, care să fie folosită în timpul mişcării:
P0: Foloseşte semnalul de salt 0
P1: Foloseşte semnalul de salt 1
P2: Foloseşte semnalul de salt 2
P3: Foloseşte semnalul de salt 3
Dacă adresa P nu este specificată, comanda numerică preia semnalul de salt 1.
G31 nu este o instrucțiune modală, ea se aplică doar în blocul în care a fost programată. Dacă instrucțiunea G31 conține o eroare de sintaxă se afişează mesajul de eroare:
3051 G22, G28, ... G31, G37.
Viteza de mişcare este:
- valoarea F specificată sau modală dacă parametrul *SKIPF*=0
- valoarea avansului preluată din *G31FD* dacă parametrul *SKIPF*=1.

În momentul în care soseşte semnalul extern, se va memora poziția axelor în variabilele de sistem specificate mai jos:
5061…..poziția axei 1
5062…..poziția axei 2
.
.
5068…..poziția axei 8

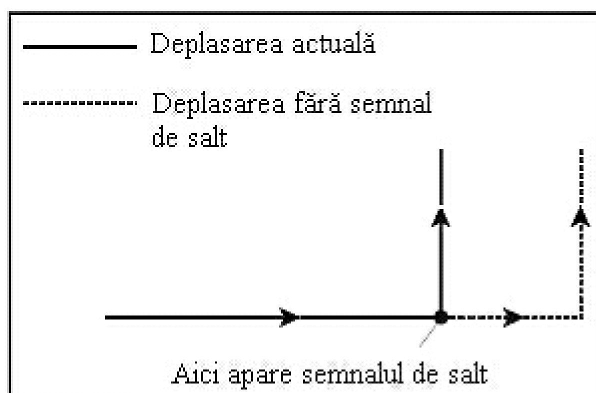


Fig. 18.1-1

Poziția memorată aici este:
- poziția din momentul în care a sosit semnalul extern,
- poziția programată a punctului final al interpolării G31 (în cazul în care nu a sosit un semnal extern),
- considerată în sistemul de coordonate al piesei curente,
- cu lungimile de compensare actuale (G43, G44) şi
- cu ofsetul actual al sculei (G45…G48) luat în considerație.

După sosirea semnalului extern, mişcarea încetineşte liniar până la oprire. Poziția finală a interpolării G31 este uşor diferită de poziția memorată în variabilele #5061… la sosirea semnalului, diferența variind funcție de avansul aplicat în interpolare. Pozițiile finale ale interpolărilor sunt accesibile în variabilele #5001… . Următoarea interpolare se va porni începând de la acele puncte finale.

Interpolarea poate fi executată doar în starea G40. Programarea G31 în starea G41 sau G42 produce afişarea mesajului de eroare *3054 G31 IN INCORRECT STATE (G31 în stare incorectă)*. Acelaşi mesaj de eroare este afişat când este activă una dintre stările G95, G51, G51.1, G68 sau G16.
Valoarea specificată la coordonatele v poate fi incrementală sau absolută. Dacă comanda de mişcare care urmează după blocul G31 este specificată în coordonate incrementale, mişcarea va fi calculată din punctul unde a sosit semnalul de salt şi s-a oprit mişcarea.
De exemplu:

```
N1 G31 G91 X100
N2 X30 Y50
```

În blocul N1 este pornită o mişcare incrementală în direcția X. Dacă comanda numerică se opreşte în punctul de coordonată X=86.7 la sosirea semnalului extern, se va mişca incremental 30 în direcția X şi 50 în direcția Z în blocul N2 (calculate din acel punct).

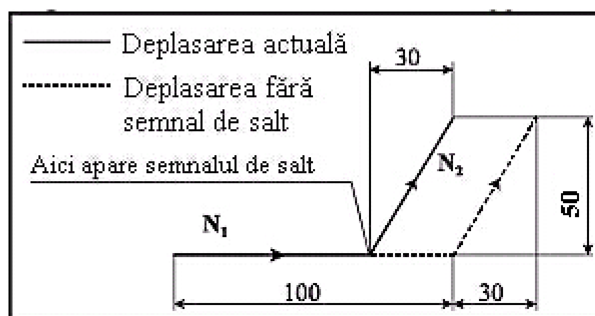


Fig. 18.1-2

În cazul în care s-a programat o specificare de date absolute, mişcarea va fi:

```
N1 G31 G90 X200
N2 X300 Y300
```

Interpolarea N1 porneşte o mişcare în direcția X către punctul de coordonată X=200. Dacă, după sosirea semnalului extern, comanda numerică se opreşte în punctul de coordonată X=167, deplasarea în direcția X va fi X=300-167, adică X=133 în blocul N2.

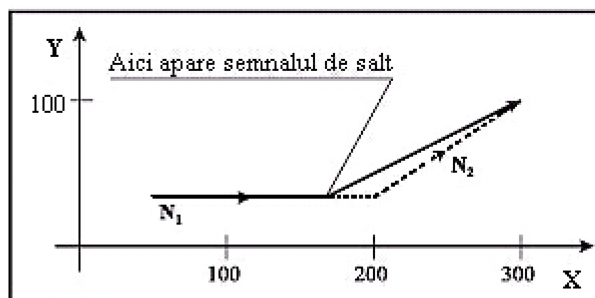


Fig. 18.1-3

18.2 Măsurarea automată a lungimii sculei (G37)

Instrucțiunea

G37 q

va determina pornirea deplasării cu avans rapid în direcția specificată la coordonata q.
Valoarea pentru q este interpretată invariabil ca o dată absolută şi este poziția prevăzută a senzorului de măsură.
Mişcarea se va executa cu avans rapid până în poziția q- *RAPDIST* unde *RAPDIST* este o valoare selectată prin parametru.
Apoi mişcarea este executată cu avansul specificat în parametrul *G37FD* până când soseşte semnalul de la sesizor sau până se afişează mesajul de eroare *3103 OUT OF RANGE (depăşire domeniu)*.
Acesta din urmă apare doar când semnalul sesizorului de atingere soseşte

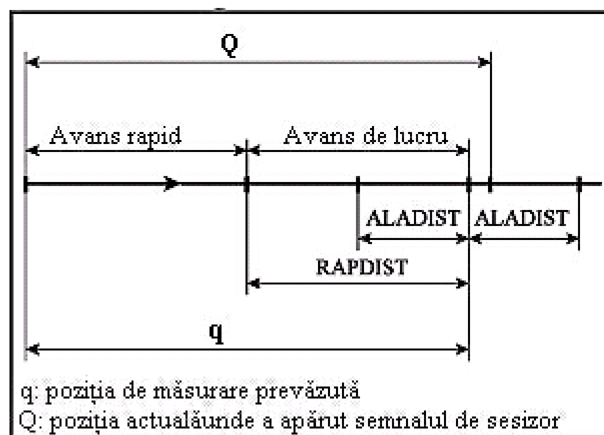


Fig. 18.2-1

în afara domeniului ALADIST (specificat în parametri) al poziției q prevăzute.
Dacă măsurătoarea este îndeplinită cu succes şi semnalul sesizorului de atingere a sosit în punctul de coordonată Q, comanda numerică va executa:
- adunarea diferenței Q-q la registrul de compensare a uzurii selectat anterior la adresa H (dacă parametrul *ADD*=1), sau
- scăderea diferenței din acesta (dacă parametrul *ADD*=0)

Anterior începerii măsurătorii trebuie impusă valoarea H corespunzătoare şi compensarea lungimii.

- G37 este o instrucțiune cu o singură acțiune.

- Ciclul G37 va fi executat în mod invariabil în sistemul de coordonate al piesei în lucru.

- Parametrii *RAPDIST* şi *ALADIST* au întotdeauna valori pozitive.
  Condiția *RAPDIST* > *ALADIST* trebuie satisfăcută pentru cei doi parametri.

- În cazul unei erori de sintaxă va fi afişat mesajul de eroare *3051 G22, G28, ... G31, G37*.

- Codul G referitor la lungimea de compensare (G43, G44, G49) nu poate fi specificat în blocul G37, altfel va fi afişat mesajul de eroare *3055 G37 IN INCORRECT STATE (G37 în stare incorectă)*.

- Acelaşi mesaj de eroare va fi afişat dacă este activă una din stările G51, G51.1, G68 sau G16.

- Mesajul de eroare *3103 OUT OF RANGE (depăşire domeniu)* este afişat pe durata execuției funcției G37 dacă semnalul sesizorului de atingere soseşte în afara domeniului *ALADIST* al poziției finale programate în interpolarea G37.

19 Funcţii de siguranţă

19.1 Verificarea programabilă a cursei (G22,G23)

Instrucţiunea

G22 X Y Z I J K P

va interzice intrarea sculei în aria interzisă. Semnificaţiile adreselor sunt:

X: Limita pe axa X în direcţia pozitivă
I: Limita pe axa X în direcţia negativă
Y: Limita pe axa Y în direcţia pozitivă
J: Limita pe axa Y în direcţia negativă
Z: Limita pe axa Z în direcţia pozitivă
K: Limita pe axa Z în direcţia negativă

Pentru datele specificate trebuie îndeplinite următoarele condiţii:

$X \geq I$, $Y \geq J$, $Z \geq K$

La adresa P se poate selecta dacă aria este interzisă în interior sau în exterior.
P=0, aria selectată este interzisă în interior.
P=1, aria selectată este interzisă în exterior.

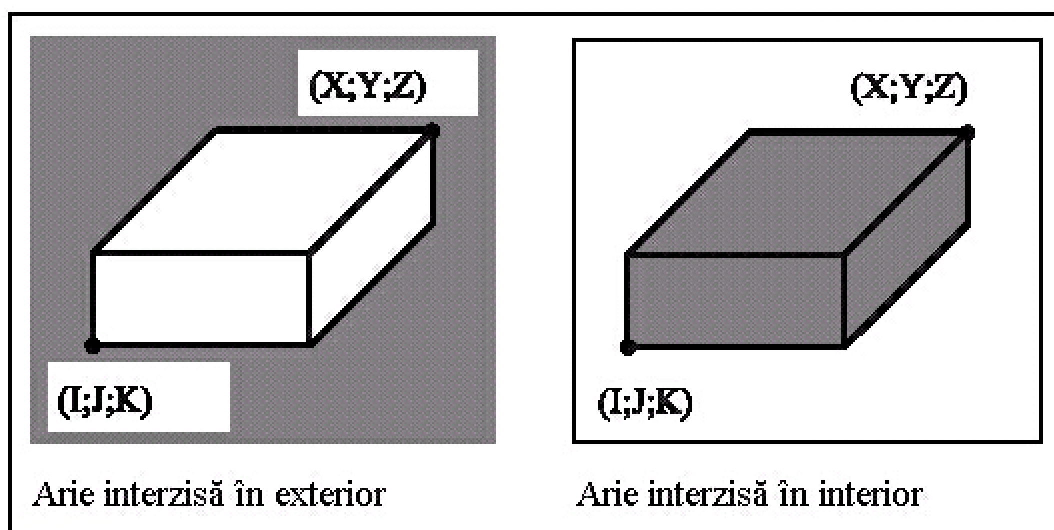


Fig. 19.1-1

Instrucţiunea

G23

va anula funcţia de verificare programabilă a cursei, scula putând să intre în aria selectată anterior. Instrucţiunile G22, G23 vor rescrie direct parametrii respectivi.
Instrucţiunile G22 şi G23 vor impune parametrul *STRKEN* la 1, respectiv la 0.
Instrucţiunile G22 P0 şi G22 P1 vor impune parametrul *EXTER* la 1, respectiv la 0.
Coordonatele X, Y, Z în instrucţiunea G22 vor scrie parametrii *LIMP2n* referitori la axele respective, coordonatele I, J, K vor impune parametrii *LIMP2n* referitori la axele respective.
Înainte de a fi înscrise la parametrii respectivi, coordonatele din instrucţiunea G22 vor fi convertite în sistemul de coordonate al maşinii, având incluse şi compensările de ofset selectate. Astfel, de exemplu, dacă este impusă compensarea de lungime pe direcţia Z atunci când este specificată instrucţiunea G22, datele limită ale coordonatelor specificate pentru

acea axă vor limita mişcarea oprind vârful sculei la limită. Dacă totuşi nu este impusă compensarea, punctul de referință al suportului sculei nu va fi admis în aria interzisă. Este recomandabil să impunem marginea ariei interzise la axa sculei, pentru cea mai lungă sculă.
- Funcția de verificare programabilă a cursei nu este disponibilă pentru axele adiționale.
- Instrucțiunile G22, G23 trebuie să fie specificate în blocuri independente.
- Funcția de verificare programabilă a cursei va fi efectivă după efectuarea întoarcerii în punctul de referință.
- Dacă maşina intră într-o arie interzisă după efectuarea întoarcerii în punctul de referință sau ca rezultat al programării G22 şi aria este interzisă la interior, interdicția trebuie să fie ridicată manual prin programarea G23; axa/axele trebuie să fie deplasate manual cu JOG şi trebuie impusă din nou verificarea cursei prin programarea G22. În cazul unei arii interzise extern, procedura de ieşire din arie va fi la fel ca aceea care se face pentru acționarea limitatorilor de fine cursă.
- Dacă pe o axă se atinge marginea ariei interzise în mişcare, se poate executa retragerea de acolo prin deplasare manuală (într-unul din modurile manuale).
- Dacă X=I, Y=J, Z=K şi E=0 atunci este disponibilă întreaga arie.
- Dacă X=I, Y=J, Z=K şi E=1 atunci este interzisă întreaga arie.
- Dacă aria este interzisă interior şi axele intră în aria interzisă sau îi ating marginea, se afişează mesajul de eroare *1400 INTERNALLY FORBIDEN AREA (arie interzisă interior).*
- Dacă aria este interzisă exterior şi axele intră în aria interzisă sau îi ating marginea, se afişează mesajul de eroare *FORBIDEN AREA t+* sau *FORBIDEN AREA t- (arie interzisă)* unde t este numele axei.

19.2 Pozițiile de sfârşit de cursă parametrice

Folosind parametrii comenzii numerice, constructorul maşinii unealtă poate defini pentru fiecare axă pozițiile de sfârşit de cursă care sunt limitele de cursă permise pentru o anumită maşină. La atingerea marginii acelei arii, comanda numerică va afişa un mesaj de eroare ca şi când ar fi călcat un limitator.
- Funcția de sfârşit de cursă parametric este executată numai după ce s-a făcut întoarcerea în punctul de referință.
- Funcția de sfârşit de cursă parametric va interzice întotdeauna o arie externă.
- Ariile de verificare programabilă a cursei şi de funcție de sfârşit de cursă se pot suprapune parțial.

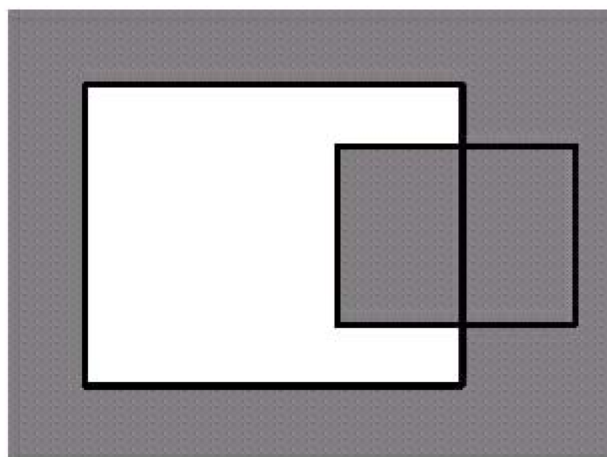


Fig. 19.2-1

19.3 Verificarea cursei înaintea mişcării

Comanda numerică diferențiază două arii interzise.
Prima este aria de sfârşit de cursă parametric care delimitează dimensiunile de mişcare fizic posibile ale maşinii. Pozițiile extreme ale acestor dimensiuni sunt denumite poziții limită. Pe durata mişcării, comanda numerică nu va permite axelor să se deplaseze dincolo de limitele ariei definite de parametri. Pozițiile limită sunt impuse de către constructorul maşinii. Utilizatorul nu poate modifica aceşti parametri.
A doua este aria definită de către funcția de verificare programabilă a cursei. Aceasta se poate face prin programarea comenzii G22 sau prin rescrierea parametrilor.
Pe durata oricăror mişcări, comanda numerică nu va permite axelor să se deplaseze dincolo de limitele acestor arii.
Dacă parametrul *CHBFMOVE* este impus 1, comanda numerică – înainte de pornirea axelor în cursul execuției unui bloc – va verifica dacă punctul final programat pentru respectiva interpolare se află în aria interzisă. Dacă punctul final al blocului se află în afara ariei de sfârşit de cursă parametric sau în interiorul ariei interzise programate, se va afişa mesajul de eroare *3056 LIMIT (limită)*, respectiv *3057 FORBIDEN AREA (arie interzisă)*. Ca rezultat nu se va începe mişcarea.

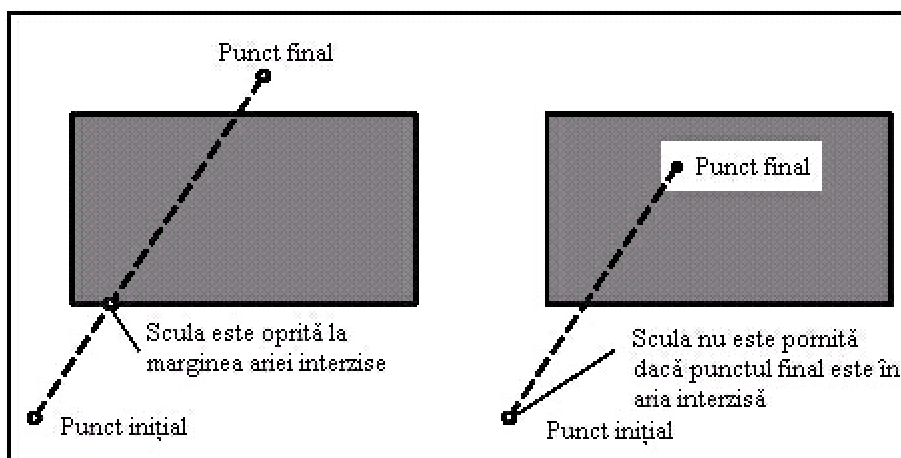


Fig. 19.3-1

Deoarece înainte de pornirea interpolării, comanda numerică verifică doar dacă punctul final al interpolării se află în interiorul ariei interzise, mesajul de eroare este produs în situațiile prezentate în figură la marginea ariei interzise, după ce s-a început mişcarea.

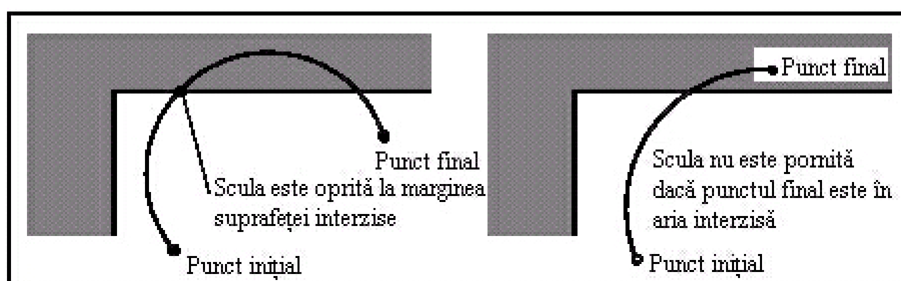


Fig. 19.3-2

20 Macro utilizator

20.1 Apelul macro simplu (G65)

Ca rezultat al instrucțiunii

G65 P(număr program) L(număr de repetări) <atribuire argument>

corpul de macro utilizator (program) specificat la adresa P (număr program) va fi apelat de atâtea ori cât este specificat la adresa L.
Corpului macro îi pot fi atribuite argumente. Acestea sunt valori numerice atribuite adreselor definite, care sunt stocate în variabilele locale respective în timpul unui apel macro. Aceste variabile locale pot fi folosite de către corpul macro, adică apelul macro este un apel de subprogram special în care programul principal poate să transfere valori (parametri) la subprogram.
Pot fi selectate următoarele două atribuiri de argumente:

<u>Șirul de adrese ale atribuirii de argumente nr.1 este</u>

A B C D E F H I J K M Q R S T U W X Y Z
La oricare dintre adresele **G, L, N, O, P** nu poate fi transferată la corpul macro nici o valoare. Adresele pot fi completate în orice secvență arbitrară, nu neapărat în ordine alfabetică.

<u>Șirul de adrese ale selectării atribuirii de argumente nr.2 este</u>

A B C I1 J1 K1 I2 J2 K2 … I10 J10 K10
În plus la adresele A, B, C, se pot atribui maxim 10 argumente diferite pentru adresele I, J, K. Adresele pot fi completate în orice secvență arbitrară. Dacă sunt selectate câteva argumente pentru o anumită adresă, variabilele vor lua valorile respective în ordinea selectării.

lv	*1. aa*	*2. aa*
#1	A	A
#2	B	B
#3	C	C
#4	I	I1
#5	J	J1
#6	K	K1
#7	D	I2
#8	E	J2
#9	F	K2
#10	(G)	I3
#11	H	J3

lv	*1. aa*	*2. aa*
#12	(L)	K3
#13	M	I4
#14	(N)	J4
#15	(O)	K4
#16	(P)	J5
#17	Q	J5
#18	R	K5
#19	S	I6
#20	T	J6
#21	U	K6
#22	V	I7

lv	*1. aa*	*2. aa*
#23	W	J7
#24	X	K7
#25	Y	I8
#26	Z	J8
#27	-	K8
#28	-	I9
#29	-	J9
#30	-	K9
#31	-	I10
#32	-	J10
#33	-	K10

- **Prescurtări**: *lv*= variabilă locală; *1.aa*=atribuirea de argumente nr.1;
2.aa=atribuirea de argumente nr.2.

Înscrierile care urmează adresele I, J, K indică secvența de atribuire a argumentelor.
Comanda numerică va accepta selectarea simultană a argumentelor 1 şi 2 într-un anumit

bloc. Se va afişa un mesaj de eroare când se încearcă să se facă referire de două ori la o variabilă cu un anumit număr. De exemplu:

G65 A2.12 B3.213 J36.9 J-12 E129.73 P2200
(1) (2) (3) (4) (5) variabila

(1) – #1=2.12
(2) – #2=3.213
(3) – #5=36.9
(4) – #8=-12
(5) – #8= eroare

În exemplul de mai sus, variabilei #8 i-a fost deja atribuită o valoare la a doua adresă J (valoarea -12), şi deoarece valoarea adresei E este de asemenea atribuită variabilei #8, se afişează mesajul de eroare *3064 BAD MACRO STATEMENT (declaraţie macro greşită)*.
La adrese se pot transfera şi un punct zecimal sau un semn.

20.2 Apel macro modal

20.2.1 Apel macro modal în fiecare comandă de mişcare (G66)

Ca rezultat al instrucţiunii:

G66 P(număr program) L(număr de repetări) <atribuire argument>

corpul de macro specificat la adresa P (număr program) va fi apelat după executarea fiecărei comenzi de mişcare, de atâtea ori cât este numărul specificat la adresa L. Interpretările adreselor P şi L şi regulile de atribuire a argumentului sunt identice cu cele descrise pentru instrucţiunea G65.
Macro selectat va fi apelat până când este anulat apelul cu comanda:

G67

Exemplu: într-un anumit segment al programului piesă trebuie executată o gaură după fiecare mişcare:
program principal:

```
...
G66 P1250 Z-100 R-1 X2 F130
G91 G0 X100------|
Y30              |
                 |-se execută găurirea după fiecare
                 | poziţionare
X150-------------|
...
G67
```

corp macro:

```
%O1250
G0 Z#18        (poziţionare cu avans rapid în direcţia Z la punctul
                specificat la adresa R-1)
G1 Z#26 F#9    (găurire până în punctul Z specificat la adresa
                Z-100, cu avansul specificat la adresa F130)
G4 P#24        (întârziere la fundul găurii cu timpul specificat la
                adresa X2)
G0 Z-[#18+#26] (retragerea sculei în punctul iniţial)
M99            (revenire în programul principal)
%
```

20.2.2 Apel macro modal din fiecare bloc (G66.1)

Ca rezultat al instrucțiunii:

G66.1 P(număr program) L(număr de repetări) <atribuire argument>

toate blocurile care urmează vor fi interpretate ca atribuiri de argumente, va fi apelat macro cu numărul specificat la adresa P şi va fi executat de atâtea ori cât este numărul specificat la adresa L.

Comanda produce acelaşi efect ca şi cum în fiecare bloc ar fi fost un apel de macro G65:

```
G66.1 P L
X Y Z---------|     |------ G65 P L X Y Z
M S           |- = -|       G65 P L M S
X ------------|     |-------G65 P L X
G67
```

Macro selectat va fi apelat până când este anulat apelul cu comanda:

G67

Regulile de atribuire a argumentului sunt:

1. În blocul care execută activarea (în care a fost programat G66.1 P L), regulile de atribuire a argumentului sunt aceleaşi ca pentru comanda G65.
2. În blocurile care urmează după instrucțiunea G66.1,
   se pot folosi aceleaşi adrese ca la comanda G65 şi
   L: #12,
   P: #16,
   G: #10 cu rezerva că comanda numerică va accepta doar o referire la o adresă G în fiecare bloc; programarea mai multor adrese va produce afişarea mesajului de eroare *3005 ILLEGAL G CODE (cod G ilegal)*.
   N: #14 dacă o adresă N este la începutul unui bloc (sau precedată cel mult de adresa unui bloc condițional„/“), a doua adresă N va fi considerată pentru argument:

```
/N130 X12.3 Y32.6 N250
 (1 )(  2 )(  3 )( 4 )      (1) numărul de bloc
                            (2) #24=12.3
                            (3) #25=32.6
                            (4) #14=250
```

   Dacă adresa N este în mijlocul blocului (precedată de oricare adresă în afară de „/“), adresa N va fi interpretată ca un argument:

```
X34.263  N320
(  1  )  (2)                (1) #24=34.236
                            (2) #14=320
```

   dacă adresa N a fost deja înregistrată ca argument, următoarea referire la adresa N va produce afişarea mesajului de eroare *3064 BAD MACRO STATEMENT (declarație macro greşită)*.

În cazul G66.1, regulile de executare a blocului sunt:

Macro selectat va fi apelat deja din blocul în care codul G66.1 a fost specificat, ținând cont de regulile de atribuire a argumentului descrise la punctul 1.
Fiecare bloc NC care urmează după G66.1 până la blocul care conține codul G67 va produce un apel macro cu regulile de atribuire a argumentului descrise la punctul 2. Nu se va face nici un apel macro dacă este găsit un bloc gol în care se face referire la o singură adresă N (de exemplu N1240), sau dintr-un bloc care conține o instrucțiune macro.

20.3 Apelul macro utilizator folosind codul G

Prin parametri se pot selecta maximum 10 coduri G diferite, prin care să fie inițiate apelurile macro. Acum în loc de specificarea

Nn G65 Pp <atribuire argument>

poate fi folosită comanda

Nn **Gg** <atribuire argument>

Numărul de program care trebuie apelat prin codul G trebuie selectat în parametri. Nici unul dintre codurile G65, G66, G66.1 şi G67 nu poate fi specificat pentru acest scop.

G(9010)=cod G de apel program O9010
G(9011)=cod G de apel program O9011
:
G(9019)=cod G de apel program O9019

Dacă în parametri este scrisă o valoare negativă, codul G selectat va genera un apel modal. De exemplu G(9011)= -120, instrucțiunea G120 din program va produce un apel modal. Starea parametrului *MODGEQU* va defini tipul de apel:

MODGEQU=0, apelul este de tip G66
MODGEQU=1, apelul este de tip G66.1

Dacă valoarea parametrului este 0, va fi apelat macro la sfârşitul fiecărui bloc de mişcare. Dacă valoarea parametrului este 1, va fi apelat macro la fiecare bloc.
Dacă este selectat un cod G standard pentru apel utilizator (de exemplu G01), şi se face o referire din nou la acel cod în corpul macro, nu se va produce un nou apel, ci va fi interpretat şi executat de către comanda numerică ca un cod G obişnuit.
Dacă se face din nou o referire la codul de apel G în corpul macro şi este diferită de un cod G standard, comanda numerică va afişa mesajul de eroare *3005 ILLEGAL G CODE (cod G ilegal)*.
- Apelul unui cod M, S, T, A, B, C utilizator dintr-un cod de apel G utilizator, sau
- apelul unui cod G utilizator dintr-un apel M, S, T, A, B, C utilizator, în funcție de valoarea parametrului:

FGMAC=0, nu e permis (se execută ca şi coduri M, S, …G obişnuite)
FGMAC=1, e permis, adică este generat un nou apel.

Codurile G utilizator au următoarele seturi de argumente:
- dacă codul este de tipul G65 sau G66, setul de argumente atribuit G65, plus P şi L,
- dacă codul este de tipul G66.1, punctele descrise sunt aplicabile pentru setul său de argumente. Un cod modal poate fi şters cu instrucțiunea G67.

20.4 Apelul macro utilizator folosind codul M

Prin parametri se pot selecta maximum 10 coduri M diferite, prin care să fie inițiate apelurile

macro. Trebuie scrisă seria de instrucțiuni:

Nn **Mm** <atribuire argument>

Codul M nu va fi transferat către PLC, ci va fi apelat macro cu numărul de program respectiv.
Numărul particular de program care să fie apelat prin apelul codului M trebuie să fie selectat prin parametri.

M(9020)=cod M de apel program O9020
M(9021)=cod M de apel program O9021
:
M(9029)=cod M de apel program O9029

Codul M poate specifica invariabil un tip de apel G65 (adică apel ne-modal).
Dacă se face referire din nou la acelaşi cod M în mijlocul corpului macro, nu se va mai apela din nou acelaşi macro, ci codul M va fi transferat la PLC.
Dacă în mijlocul corpului macro se face un apel utilizator de tipul G, S, T, A, B, C sau de tip M, atunci pentru

FGMAC=0, apelul nu este permis (se execută codurile simple M, S, … G)
FGMAC=1, apelul este permis, adică se generează un apel nou.

Un cod M selectat prin parametri pentru a iniția un apel macro poate fi precedat în bloc doar de „/" şi de adresa N.
Un bloc care conține un apel macro inițiat printr-un cod M poate să includă doar un singur cod M.
Setul de argumente nr.1:

A B C D E F G H I J K L P Q R S T U V W X Y Z

Setul de argumente nr.2 se poate folosi de asemenea cu funcția M.

20.5 Apelul de subprogram cu codul M

Prin parametri se pot selecta maximum 10 coduri M, prin care se pot iniția apelurile de subprogram. În locul instrucțiunii

Nn Gg Xx Yy M98 Pp

Se poate specifica

Nn Gg Xx Yy **Mm**

Codul M selectat nu va fi transferat la PLC, ci va fi apelat subprogramul respectiv.
Numărul particular de program care să fie apelat prin codul M poate fi selectat prin următorii parametri:

M(9000)=cod M de apel program O9000
M(9001)=cod M de apel program O9001
:
M(9009)=cod M de apel program O9009

Dacă se face referire din nou la acelaşi cod M în subprogram, acesta din urmă nu va mai apela din nou acelaşi subprogram, ci codul M va fi transferat la PLC.
Dacă în subprogram se face un apel utilizator de tipul G, S, T, A, B, C sau de tip M, atunci pentru

FGMAC=0, apelul nu este permis (se execută codurile simple M, S, … G)
FGMAC=1, apelul este permis, adică se generează un apel nou.

20.6 Apelul de subprogram cu codul T

Impunând parametrul T(9034)=1, valoarea lui T scrisă în program nu va fi transferată către PLC, ci codul T va iniția apelul subprogramului cu numărul O9034.
Blocul nou

Gg Xx Yx **Tt**

va fi echivalent cu următoarele două blocuri:

#199=t
Gg Xx Yy M98 P9034

Valoarea atribuită la adresa T va fi transferată ca un argument la variabila comună #199.
Dacă în subprogramul pornit cu codul T se face referire din nou la adresa T, subprogramul nu va fi apelat din nou, ci valoarea adresei T va fi transferată la PLC.
Dacă în subprogram se face un apel utilizator de tipul G, M, S, A, B, C, atunci pentru

FGMAC=0, apelul nu este permis (se execută codurile simple M, S, … G)
FGMAC=1, apelul este permis, adică se generează un apel nou.

20.7 Apelul de subprogram cu codul S

Impunând parametrul S(9033)=1, valoarea lui S scrisă în program nu va fi transferată către PLC, ci codul S va iniția apelul subprogramului cu numărul O9033.
Blocul nou

Gg Xx Yx **Ss**

va fi echivalent cu următoarele două blocuri:

#198=s
Gg Xx Yy M98 P9033

Valoarea atribuită la adresa S va fi transferată ca un argument la variabila comună #198.
Dacă în subprogramul pornit cu codul S se face referire din nou la adresa S, subprogramul nu va fi apelat din nou, ci valoarea adresei va fi transferată la PLC.
Dacă în subprogram se face un apel utilizator de tipul G, M, T, A, B, C, atunci pentru

FGMAC=0, apelul nu este permis (se execută codurile simple M, S, … G)
FGMAC=1, apelul este permis, adică se generează un apel nou.

20.8 Apelul de subprogram cu codurile A, B, C

Dacă adresa A, B sau C este definită ca o funcție auxiliară prin parametri (1493 A.MISCEL=1, 1496 B.MISCEL=1, 1499 C.MISCEL=1) şi este impus parametrul A(9030)=1, B(9031)=1 sau C(9032)=1, valoarea lui A, B sau C scrisă în program nu va fi transferată către PLC sau interpolator, ci codul A, B sau C va iniția apelul subprogramului corespunzător nr.O9030, O9031 sau O9032.
Blocul

Gg Xx Yy **Bb**

este echivalent cu următoarele două blocuri:

#196=b
Gg Xx Yy M98 P9031

Valorile atribuite la adresele **A**, **B** şi **C** vor fi transferate la variabilele comune **#195**, **#196** şi respectiv **#197**.

Dacă se face referire din nou la aceeaşi adresă în subprogramul pornit cu codul A, B sau C, subprogramul nu va fi apelat din nou, ci valoarea adresei va fi transferată către PLC sau interpolator.
Dacă în subprogram se face un apel utilizator de tipul G, M, S, T atunci pentru
FGMAC=0, apelul nu este permis (se execută codurile simple M, S, … G)
FGMAC=1, apelul este permis, adică se generează un apel nou.

20.9 Diferențe între apelul unui subprogram şi apelul unui macro

- Apelul de macro poate include atribute, cel de subprogram nu.
- Apelul unui subprogram se poate ramifica în subprogram numai după executarea celorlalte comenzi programate în bloc; un apel de macro doar se va ramifica.
- Apelul de macro modifică nivelurile variabilelor locale, pe când un apel de subprogram nu. De exemplu valoarea #1 înainte de apelul G65 este diferită de cea din mijlocul corpului macro. Valoarea #1 înainte de M98 este identică cu cea din subprogram.

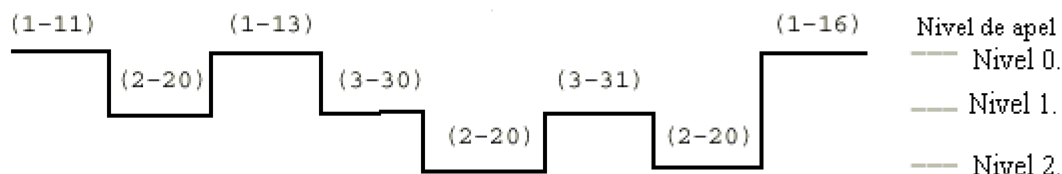
20.9.1 Apeluri multiple

Dintr-un macro se poate apela un alt macro. Apelurile de macro se pot face în patru nivele de ramificare, incluzând apeluri simple şi modale. Incluzând şi apelurile de subprograme, nivelul maxim de ramificare a apelurilor poate fi de 8 nivele.
În cazul apelurilor multiple de macro modale (tip G66), după executarea fiecărui bloc de interpolare se va apela la început ultimul macro specificat, din care va fi apelat macro specificat anterior într-o secvență care merge înapoi. Să examinăm exemplul următor:

```
%O0001
...
N10 G66 P2
N11 G1 G91 Z10   (1-11)
N12 G66 P3
N13 Z20          (1-13)
N14 G67          (anularea apelului G66 P3...)
N15 G67          (anularea apelului G66 P2 ...)
N16 Z-5          (1-16)
...
%O0002
N20 X4           (2-20)
N21 M99
%

%O0003
N30 Z2           (3-30)
N31 Z3           (3-31)
N32 M99
%
```

Incluzând doar interpolările, secvenţa de executare va fi:



Primul număr din paranteză reprezintă numărul programului iar al doilea numărul blocului executat.
Instrucţiunea G67 specificată în blocul N14 va anula apelul de macro din blocul N12 (O0003); cea specificată în blocul N15 va anula apelul de macro din blocul N10 (O0002).
În cazul apelurilor multiple de macro tip G66.1, primul va fi apelat ultimul macro specificat la începutul fiecărui bloc (considerând adresele blocului respectiv ca argumente), apoi va fi apelat macro specificat anterior, introducând blocurile acelui macro şi considerându-le ca argumente.
Dacă un alt macro este apelat din nou din macro, nivelele variabilelor locale vor creşte odată cu nivelele de macro.

program principal	macro	macro	macro	macro
nivel 0	nivel 1	nivel 2	nivel 3	nivel 4
	O____	O____	O____	O____
G65 P	G65 P	G65 P	G65 P	
	M99	M99	M99	M99
variabile locale				
nivel 0	nivel 1	nivel 2	nivel 3	nivel 4
#1	#1	#1	#1	#1
	:	:	:	:
#33	#33	#33	#33	#33

Când este apelat primul macro, variabilele locale ale programului principal vor fi stocate (de la #1 la #33), şi variabilele locale de nivel 1 vor lua valorile argumentelor specificate în apel. Dacă este apelat un alt macro de pe primul nivel, variabilele locale ale primului nivel vor fi stocate (de la #1 la #33), şi variabilele locale de nivel 2 vor lua valorile argumentelor specificate în apel. În cazul apelurilor multiple, variabilele locale ale nivelului anterior vor fi stocate şi variabilele locale ale următorului nivel vor lua valorile argumentelor specificate în apel. În cazul lui M99, revenirea din macro apelat în programul care a făcut apelul, variabilele locale stocate pe nivelul anterior vor fi restaurate în aceleaşi stări în care erau în momentul în care au fost stocate pe timpul apelului.

20.10 Formatul corpului macro utilizator

Formatul de program al unui macro utilizator este identic cu cel al unui subprogram:

O (număr program)
:
comenzi
:
M99

Numărul de program este fără importanță, dar numerele de program cuprinse între O9000 şi O9034 sunt rezervate pentru apeluri speciale.

20.11 Variabile ale limbajului de programare

În programele principale, subprograme şi macro, pot fi atribuite adreselor variabile în locul valorilor numerice specifice. O valoare poate fi atribuită pentru fiecare variabilă în interiorul domeniului permis. Folosirea variabilelor va permite proceduri de programare mult mai flexibile.
O anumită dată poate fi transformată în parametru prin folosirea variabilelor comune în programele principale şi în subprograme, astfel nu va mai fi necesar să se scrie programe noi pentru piese similare dar cu dimensiuni diferite. Pentru aceasta operatorul va scrie din nou doar variabilele comune corespunzătoare .
Folosirea variabilelor poate face un macro mult mai flexibil decât un subprogram convențional. În timp ce argumentele nu pot fi transferate la un subprogram, ele pot fi ataşate la un macro prin intermediul variabilelor locale.

20.11.1 Identificarea unei variabile

Se poate folosi un număr de variabilă astfel ca fiecare variabilă să fie identificată prin numărul ei. O variabilă este compusă din codul # şi un număr. De exemplu:

#12
#138
#5106

Se poate folosi de asemenea o formulă pentru a se face referire la o formulă - #[<formulă>].
De exemplu:

#[#120] înseamnă că variabila 120 conține numărul de serie al variabilei la care se face referire;
#[#120-4] înseamnă că numărul de variabilă referit este obținut scăzând 4 din numarul conținut în variabilă.

20.11.2 Referirea la o variabilă

Adresele diferite din cuvintele unui bloc de program pot lua valori de variabile la fel ca şi valori numerice. Semnul minus („-“) sau operatorul I poate fi folosit, atunci când este permis pentru valori numerice, şi atunci când se face referire la o variabilă după o adresă.
De exemplu:

G#102 dacă #102=1.0, această referire este echivalentă cu G1
XI-#24 dacă #24=135.342, această referire este echivalentă cu XI-135.342

- Nu este permisă referirea la un număr de program O, număr de bloc N sau bloc condițional

/ printr-o variabilă. Adresa N va fi privită ca un număr de bloc dacă este precedată doar de adresa „/“ în bloc.

- Numărul unei variabile nu poate fi înlocuit printr-o variabilă, adică ##120 nu este permis. Specificarea corectă este #[#120].
- Dacă variabila este folosită după o adresă, valoarea ei nu poate depăşi domeniul de valori admisibil pentru respectiva adresă. Dacă de exemplu #112=5630, referirea M#112 va produce un mesaj de eroare.
- Dacă variabila este folosită după o adresă, valoarea ei va fi rotunjită la o cifră semnificativă corespunzătoare adresei. De exemplu:

```
M#112     va fi  M1     pentru #112=1.23
M#112     va fi  M2     pentru #112=1.6
```

20.11.3 Variabile vacante

O variabilă la care nu s-a făcut referire (nedefinită) este vacantă. Variabila #0 este folosită pentru o variabilă care este vacantă întotdeauna:

```
#0=<vacantă>
```

20.11.4 Formatul numeric al variabilelor

Fiecare variabilă este reprezentată prin 32 de biţi de mantisă şi 8 biţi de caracteristică.

variabila= $M*2^C$

Reprezentarea unei variabile **vacante**, M=0, C=0

Reprezentarea unei variabile **0 - valoare**, M=0, C= -128

Natura unei variabile vacante, comparate într-o adresă:

Referire la o variabilă **vacantă** într-o adresă:

```
Dacă #1=<vacantă>             dacă #1=0

G90 X20 Y#1                   G90 X20 Y#1
        |                             |
   G90 X20                    G90 X20 Y0
```

Variabilă **vacantă** într-o instrucţiune *definiţie*:

```
dacă #1=<vacantă>             dacă #1=0

#2=#1                         #2=#1
   |                             |
 #2=<vacantă>                  #2=0

#2=#1*3                       #2=#1*3
   |                             |
 #2=0                          #2=0

#2=#1+#1                      #2=#1+#1
   |                             |
 #2=0                          #2=0
```

Diferența între o variabilă **vacantă** şi una cu **valoare 0** într-o **expresie condițională** va fi:

dacă #1=<vacantă>	dacă #1=0
#1 EQ #0	#1 EQ #0
|	|
îndeplinită	neîndeplinită
#1 NE 0	#1 NE 0
|	|
îndeplinită	neîndeplinită
#1 GE 0	#1 GE 0
|	|
îndeplinită	neîndeplinită
#1 GT 0	#1 GT 0
|	|
îndeplinită	neîndeplinită

20.12 Tipuri de variabile

Variabilele sunt clasificate în funcție de proprietățile şi modul lor de folosire în: locale, comune şi sistem. Numărul de ordine al variabilelor defineşte categoria la care aparțin.

20.12.1 Variabile locale (de la #1 la #33)

Variabila locală este o variabilă folosită de către programul macro local. Dacă macro A apelează B, şi este făcută referire la variabila locală #i în fiecare din macro-urile A şi B, valoarea variabilei locale #i la nivelul macro A nu se va pierde şi nu va fi rescrisă după ce a fost apelat macro B – cu toate că se face referire la #i şi în macro B. Variabilele locale sunt folosite pentru transferul argumentelor. Împerecherile dintre adresele argumentelor şi variabilele locale sunt conținute în tabelul din capitolul care descrie procedura de apel simplu de macro (G65).
Variabila locală a cărei adresă nu a fost implicată în atribuirea de argumente, este una vacantă care poate fi folosită după dorință.

20.12.2 Variabile comune (de la #100 la #199, de la #500 la #599)

Spre deosebire de variabilele locale, variabilele comune sunt identice pe parcursul întregului program (nu doar la un nivel al unui apel de program) – indiferent dacă sunt în programul principal, în subprogram sau într-un macro, sau la oricare nivel al macro. Prin urmare, dacă #i a fost utilizat într-un macro, de exemplu i-a fost atribuită o valoare, #i va avea aceeaşi valoare într-un alt macro, până când va fi rescrisă. Variabilele comune pot fi utilizate în mod liber în sistem, ele nu au funcții distinctive.

Variabilele comune de la #100 la #199 vor fi şterse la deconectarea alimentării.
Valorile variabilelor comune de la #500 la #599 se vor păstra chiar şi după o deconectare a alimentării.
Variabilele macro de la #500 la #599 se pot face „protejate la scriere“ prin intermediul parametrilor *WPROT1* şi *WPROT2* . Numărul primului element al blocului care trebuie protejat va fi scris la parametrul *WPROT1* iar al ultimului la *WPROT2* .

Dacă de exemplu trebuie protejate variabilele de la #530 la #540, trebuie impuşi parametrii *WPROT1*=530 şi *WPROT2*=540.

20.12. 3 Variabile de sistem

Variabilele de sistem sunt fixate si asigură informație despre stările sistemului.

Semnale de intrare interfață - #1000-#1015, #1032

Pot fi determinate 16 semnale de intrare interfață, unul câte unul, prin citirea variabilelor de sistem de la #1000 la #1015.

Numele variabilelor de sistem	Intrare interfață cu referire la programul PLC
#1000	I[CONST+000]
#1001	I[CONST+001]
#1002	I[CONST+002]
#1003	I[CONST+003]
#1004	I[CONST+004]
#1005	I[CONST+005]
#1006	I[CONST+006]
#1007	I[CONST+007]
#1008	I[CONST+010]
#1009	I[CONST+011]
#1010	I[CONST+012]
#1011	I[CONST+013]
#1012	I[CONST+014]
#1013	I[CONST+015]
#1014	I[CONST+016]
#1015	I[CONST+017]

unde CONST=I_LINE*10 şi *I_LINE* este un parametru. Deci se poate citi oricare intrare interfață.

Valorile variabilelor de mai sus sunt:

0= dacă contactul de la intrare este deschis,

1= dacă contactul de la intrare este închis.

Cele 16 intrări de mai sus se pot citi simultan la variabila #1032. Funcție de atribuirea variabilelor de sistem pentru citirea una câte una, valoarea va fi:

$$\#1032=\sum_{i=0}^{15}\#[1000+i]*2^i$$

Prin urmare, cu 24V aplicați la intrările de la #1002 la #1010, restul intrărilor fiind deschise, valoarea variabilei #1032 va fi:

$$\#1032 = 1*2^2 + 1*2^{10} = 1028$$

Variabilele intrărilor interfață sunt „numai citire“, şi nu pot fi folosite în partea stângă a unei instrucțiuni definiție.

Semnale de ieşire interfaţă - #1100-#1115, #1132

Pot fi transmise 16 semnale de ieşire interfaţă, unul câte unul, prin atribuirea valorilor variabilelor de la #1100 la #1115.

Numele variabilelor de sistem	Ieşire interfaţă cu referire la programul PLC
#1100	Y[CONST+000]
#1101	Y[CONST+001]
#1102	Y[CONST+002]
#1103	Y[CONST+003]
#1104	Y[CONST+004]
#1105	Y[CONST+005]
#1106	Y[CONST+006]
#1107	Y[CONST+007]
#1108	Y[CONST+010]
#1109	Y[CONST+011]
#1110	Y[CONST+012]
#1111	Y[CONST+013]
#1112	Y[CONST+014]
#1113	Y[CONST+015]
#1114	Y[CONST+016]
#1115	Y[CONST+017]

unde CONST=O_LINE*10 iar *O_LINE* este un parametru. Deci se poate transmite sau citi oricare cuvânt ieşire interfaţă.
Valorile variabilelor de mai sus pot fi:

0= dacă contactul de la ieşire este deschis,
1= dacă contactul de la ieşire este închis.

Cele 16 ieşiri de mai sus se pot citi simultan la variabila #1132. Funcţie de atribuirea variabilelor de sistem pentru fiecare ieşire, valoarea va fi:

$$\#1132=\sum_{i=0}^{15}\#[1100+i]*2^{i}$$

Prin urmare, cu ieşirile de la #1102 la #1109 active, restul ieşirilor fiind inactive, valoarea variabilei #1132 va fi:

$$\#1132 = 1*2^{2} + 1*2^{9} = 516$$

Valorile de compensare pentru sculă – de la #10001 la #13999

Valorile de compensare pentru sculă pot fi citite la variabilele de la #10001 la #13999, sau li se pot atribui valori.

Nr. compensare	H		D	
	geometrie	uzură	geometrie	uzură
1	#10001	#11001	#12001	#13001
2	#10002	#11002	#12002	#13002
.	.	.	.	.
.	.	.	.	.
999	#10999	#11999	#12999	#13999

Ofseturile punctelor de zero ale piesei – de la #5201 la #5328

Ofseturile punctelor de zero ale piesei pot fi citite la variabilele de la #5201 la #5328, sau li se pot atribui valori.

Numărul variabilei	valoarea variabilei	sistemul de coordonate al piesei
#5201	ofset comun al punctului de zero al piesei, axa 1	comun pentru toate sistemele de coordonate
#5202	ofset comun al punctului de zero al piesei, axa 2	
:		
#5208	ofset comun al punctului de zero al piesei, axa 8	
#5221	valoarea ofsetului punctului de zero al piesei, axa 1	G54
#5222	valoarea ofsetului punctului de zero al piesei, axa 2	
:		
#5228	valoarea ofsetului punctului de zero al piesei, axa 8	
#5241	valoarea ofsetului punctului de zero al piesei, axa 1	G55
#5242	valoarea ofsetului punctului de zero al piesei, axa 2	
:		
#5248	valoarea ofsetului punctului de zero al piesei, axa 8	
#5261	valoarea ofsetului punctului de zero al piesei, axa 1	G56
#5262	valoarea ofsetului punctului de zero al piesei, axa 2	
:		
#5268	valoarea ofsetului punctului de zero al piesei, axa 8	
#5281	valoarea ofsetului punctului de zero al piesei, axa 1	G57
#5282	valoarea ofsetului punctului de zero al piesei, axa 2	
:		
#5288	valoarea ofsetului punctului de zero al piesei, axa 8	
#5301	valoarea ofsetului punctului de zero al piesei, axa 1	G58
#5302	valoarea ofsetului punctului de zero al piesei, axa 2	
:		
#5308	valoarea ofsetului punctului de zero al piesei, axa 8	
#5321	valoarea ofsetului punctului de zero al piesei, axa 1	G59
#5322	valoarea ofsetului punctului de zero al piesei, axa 2	
:		
#5368	valoarea ofsetului punctului de zero al piesei, axa 8	

Numărul axei se referă la axele fizice. Relația dintre numerele şi numele axelor va fi definită de către constructorul maşinii unealtă prin parametrii din grupul *AXIS*. De obicei axele

1,2 şi 3 sunt atribuite la adresele X, Y şi respectiv Z, dar sunt permise şi alte specificaţii.

Alarma - #3000

Prin definirea

#3000=nnn(ALARMĂ)

se poate afişa un mesaj numeric de eroare (nnn= maxim trei cifre zecimale) şi textul mesajului de eroare. Textul trebuie pus între paranteze rotunde ( ). Mesajul nu poate fi mai lung de 25 de caractere. Dacă macro conţine o eroare, adică programul rulează către o ramură în care o valoare a fost definită la variabila #3000, programul va fi executat până la blocul anterior, apoi se suspendă execuţia şi sunt afişate pe ecran un mesaj de eroare şi codul său (4nnn). Numărul mesajului este suma numărului specificat în variabila #3000 şi 4000. Dacă nu a fost specificat nici un număr, codul mesajului va fi 4000 iar dacă nu a fost specificat nici un text, câmpul mesajului este gol. Starea de eroare se poate şterge cu butonul RESET.

Cronometru (timer) în milisecunde - #3001

Valoarea variabilei #3001 se poate citi şi înscrie.
Intervalul dintre două momente de timp poate fi măsurat în milisecunde, cu o acurateţe de circa 20 milisecunde. Numărătorul #3001 se va umple la 65536. Valoarea variabilei #3001 va începe de la 0 în momentul cuplării la reţea, şi va număra crescător. Numărarea continuă cât timp este pornită comanda numerică.

Cronometru (timer) principal - #3002

Valoarea variabilei #3002 se poate citi şi înscrie.
Intervalul dintre două momente de timp poate fi măsurat în milisecunde, cu o acurateţe de circa 20 milisecunde.
În momentul cuplării la reţea, valoarea variabilei #3002 va fi cea din momentul decuplării, şi va număra crescător.
Numărarea se face atât timp cât luminează START, adică este măsurat cât timp stă comanda numerică în condiţia de start. Este localizat la ceasul *CUTTING2* din memoria de parametri.

Suprimarea execuţiei bloc cu bloc - #3003

Dacă #3003=1, comanda numerică nu se va opri la sfârşitul unui bloc (în starea de mod bloc singular) până când variabila nu ia valoarea 0.
Valoarea variabilei este 0 la deconectarea alimentării sau după resetarea programului la început.

#3003 execuţie bloc cu bloc
0= nesuprimat
1= suprimat

Suprimarea butonului de stop, a corecției avansului şi a stopului precis - #3004

În condiția de suprimare a funcției de oprire a avansului, avansul se va opri la apăsarea butonului de stop doar când se înlătură suprimarea.
Când este suprimată corecția avansului, corecția ia valoarea de 100% până când se înlătură suprimarea.
În condiția de suprimare a stopului precis, comanda numerică nu va executa verificarea până când nu se înlătură suprimarea.
Valoarea variabilei este 0 la conectarea alimentării sau după resetarea programului la început.

#3004	Stop precis	Corecție avans	Stop avans
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

0 = funcția este activă
1 = funcția este suprimată

Stop cu mesaj - #3006

Ca rezultat al atribuirii unei valori la
#3006=nnn(MESAJ)
execuția programului este oprită şi este afişat pe ecran mesajul între paranteze rotunde împreună cu codul 5nnn. Codul este suma numărului specificat în variabilă şi 5000. Dacă nu a fost specificat nici un număr, va fi afişat codul 5000, iar dacă nu a fost specificat nici un text, câmpul mesajului va fi gol. Execuția programului se reia la eliberarea butonului START, iar mesajul este şters de pe ecran. Mesajul nu poate fi mai lung de 25 de caractere. Această instrucțiune este utilă oricând este necesară intervenția operatorului pe timpul execuției programului.

Starea imaginii în oglindă - #3007

Prin citirea variabilei #3007, operatorul poate stabili axa fizică particulară la care s-a înregistrat comanda de imagine în oglindă. Această variabilă este de tipul „doar citire”.
Valoarea variabilei este interpretată în termeni binari după cum urmează:

```
1 1 1 1 1 1
5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0        (1) prima axă
                |         | | |        (2) a doua axă
               (8)       (3)2)1)       (3) a treia axă
                                         :
                                       ( 8) a opta axă
```

Biții au următoarea semnificație
0= fără imagine în oglindă
1= imagine în oglindă
Dacă de exemplu valoarea variabilei este 5, imaginea în oglindă este activă pe axele 1 şi 3

Numărul axei se referă la o axă fizică, parametrul definind numele particular al axei referitor la un număr de axă fizică.

Numărul de piese prelucrate, numărul de piese de prelucrat - #3901, #3902

Comanda numerică memorează numărul de piese prelucrate în numărătorul #3901. Conținutul numărătorului va fi incrementat cu unu după executarea fiecărei funcții M02, M30 sau a funcțiilor M selectate în parametrul *PRTCNTM*. Când numărul de piese prelucrate devine egal cu numărul necesar de piese (numărătorul #3902), comanda numerică semnalizează către PLC printr-un flag.

Numărul de piese prelucrate #3901
Numărul de piese de prelucrat #3902

Numărătoarele #3901 şi #3902 se află la parametrii *PRTTOTAL* şi respectiv *PRTREQRD*.

Informație modală – de la #4001 la #4130, şi de la #4201 la #4330

Valorile modale eficace în blocul anterior pot fi stabilite prin citirea variabilelor de sistem de la #4001 la #4130.
Comenzile modale active în blocul aflat în execuție pot fi stabilite prin citirea variabilelor de la #4201 la #4330.

variabila de sistem	informația modală a blocului anterior	variabila de sistem	informația modală a blocului aflat în execuție
#4001	cod G, grup 1	#4201	cod G, grup 1
:	:	:	:
#4020	cod G, grup 20	#4220	cod G, grup 20
#4101	cod A	#4301	cod A
#4102	cod B	#4302	cod B
#4103	cod C	#4303	cod C
#4107	cod D	#4307	cod D
#4108	cod E	#4308	cod E
#4109	cod F	#4309	cod F
#4111	cod H	#4311	cod H
#4113	primul cod M introdus	#4313	primul cod M introdus
#4114	număr bloc, N	#4314	număr bloc, N
#4115	număr program, O	#4315	număr program, O
#4119	cod S	#4319	cod S
#4120	cod T	#4320	cod T

Informație pozițională

Poziții la sfârşit de bloc

Variabilă de sistem	informație de poziție	citire pe durata mişcării
#5001	coordonata pe axa 1 la sfârşitul blocului	
#5002	coordonata pe axa 2 la sfârşitul blocului	
:		posibilă
#5008	coordonata pe axa 8 la sfârşitul blocului	

Coordonata la sfârşitul blocului va fi introdusă în variabilă
- în sistemul de coordonate al piesei în lucru
- ținând cont de ofseturile coordonatei
- în coordonate carteziene
- cu toate compensările ignorate (lungime, rază, ofset sculă)

Pozițiile instantanee în sistemul de coordonate al maşinii

Variabilă de sistem	natura informației de poziție	introducere pe durata mişcării
#5021	coordonata instantanee pe axa 1 (G53)	
#5022	coordonata instantanee pe axa 2 (G53)	
:		nu e posibilă
#5028	coordonata instantanee pe axa 8 (G53)	

Poziția instantanee (G53) va fi introdusă în variabilă
- în sistemul de coordonate al maşinii
- ținând cont de toate compensările (lungime, rază, ofset sculă).

Pozițiile instantanee în sistemul de coordonate al piesei

Variabilă de sistem	natura informației de poziție	introducere pe durata mişcării
#5041	coordonata instantanee pe axa 1	
#5042	coordonata instantanee pe axa 2	
:		nu e posibilă
#5048	coordonata instantanee pe axa 8	

Coordonata instantanee va fi introdusă în variabilă
- în sistemul de coordonate al piesei în lucru
- ținând cont de ofseturile coordonatei
- în coordonate carteziene
- ținând cont de toate compensările (lungime, rază, ofset sculă).

Poziția in care apare semnalul de salt

Variabilă de sistem	natura informației de poziție	introducere pe durata mişcării
#5061	coordonata semnalului de salt pe axa 1 (G31)	
#5062	coordonata semnalului de salt pe axa 2 (G31)	
:		posibilă
#5068	coordonata semnalului de salt pe axa 8 (G31)	

Poziția în care a sosit semnalul de salt în blocul G31 va fi introdusă în variabilă
- în sistemul de coordonate al piesei
- ținând cont de ofseturile coordonatei
- în coordonate carteziene
- cu toate compensările (lungime, rază, ofset sculă).

Dacă semnalul de salt nu soseşte, variabilele de mai sus vor lua poziția finală programată în blocul G31.

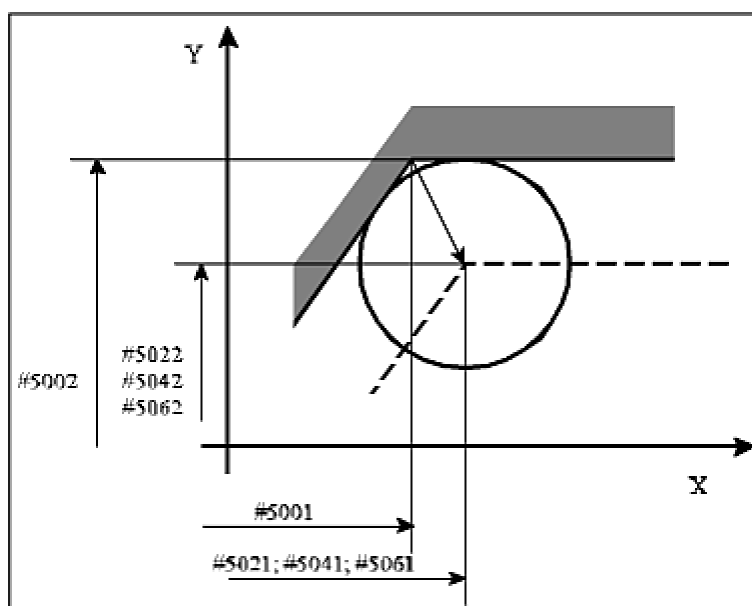


Fig. 20.12.3-1

Compensarea lungimii sculei

Variabilă de sistem	natura informației de poziție	introducere pe durata mişcării
#5081	lungimea compensării pe axa 1	
#5082	lungimea compensării pe axa 2	
:		nu e posibilă
#5088	lungimea compensării pe axa 8	

Valoarea care poate fi citită a compensării lungimii sculei este cea care este activă în blocul aflat în execuție.

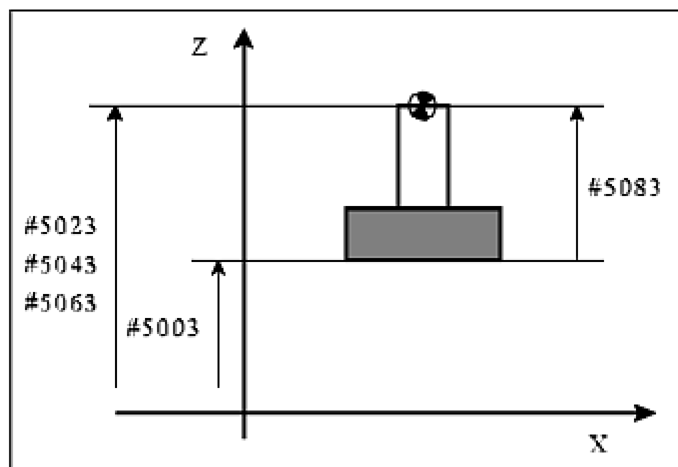


Fig. 20.12.3-2

Decalajul Servo

Variabilă de sistem	natura informației de poziție	introducere pe durata mişcării
#5101	decalaj servo pe axa 1	
#5102	decalaj servo pe axa 2	
:		nu e posibilă
#5108	decalaj servo pe axa 8	

Valoarea care poate fi citită a decalajului servo este o valoare cu semn, în milimetri.

20.13 Instrucțiuni ale limbajului de programare

Expresia

#i = <formulă>

este folosită pentru descrierea diferitelor instrucțiuni. Expresia <formulă> poate să includă operații aritmetice, funcții, variabile sau constante.
În general, referirile în <formulă> se fac la variabilele #i şi #k.
Termenul <formulă> nu e obligatoriu să se afle în partea dreaptă a unei instrucțiuni definiție, se poate atribui o formulă şi pentru diferitele adrese în blocul NC, în locul unei anumite valori numerice sau variabile.

20.13.1 Definire, Înlocuire: #i =#j

Codul instrucțiunii este =.
Ca rezultat al instrucțiunii, variabila #i va lua valoarea variabilei #j, adică valoarea variabilei #j va fi introdusă în variabila #i.

20.13.2 Operații aritmetice şi funcții

Operații cu un singur operand

Minus cu un singur operand: #i = - #j

Codul operației este -.

Ca rezultat al operației, variabila #i va avea o valoare identică cu variabila #j în valoare absolută dar cu semn schimbat.

Negație aritmetică: #i = NOT #j

Codul operației este **NOT**.

Ca rezultat al operației, variabila #j este convertită mai întâi într-un număr de 32 de biți cu virgulă fixă. Se afişează mesajul de eroare *3091 ERRONEUS OPERATION WITH # (operare incorectă cu #)* dacă numărul nu poate fi reprezentat în 32 de biți. Apoi valoarea numărului binar va fi negată bit cu bit şi numărul rezultat astfel va fi reconvertit într-unul cu virgulă mobilă şi pus în variabila #i.

Operații aritmetice de sumare

Adunarea: #i = #j + #k

Codul operației este +.

Ca rezultat al operației, variabila #i va deveni suma dintre valoarea variabilei #j şi valoarea variabilei #k.

Scăderea: #i = #j - #k

Codul operației este -.

Ca rezultat al operației, variabila #i va deveni diferența dintre valoarea variabilei #j şi valoarea variabilei #k.

Suma logică, SAU: #i = #j OR #k

Codul operației este **OR (SAU logic)**.

Ca rezultat al operației, suma logică dintre variabilele #j şi #k se va introduce în variabila #i pentru fiecare din cei 32 de biți. Se va introduce 0 atunci când pozițiile similare din cele două numere au valoarea 0 (altfel se introduce 1).

SAU exclusiv: #i = #j XOR #k

Codul operației este **XOR (SAU exclusiv)**.

Ca rezultat al operației, variabilele #j şi #k vor fi adunate pentru fiecare din cei 32 de biți în variabila #i de o aşa manieră că se va introduce 0 atunci când pozițiile similare din cele două numere au valori identice (şi 1 când valorile sunt diferite).

Operații aritmetice de multiplicare

Înmulțirea: #i = #j * #k

Codul operației este *.

Ca rezultat al operației, variabila #i va deveni produsul dintre valoarea variabilei #j şi valoarea variabilei #k.

Împărțirea: #i = #j / #k

Codul operației este /.

Ca rezultat al operației, variabila #i va deveni raportul variabilelor #j şi #k. Valoarea lui #k nu poate fi 0, pentru că dacă este 0 se afişează mesajul de eroare *3092 DIVISION BY 0 # (împărțire prin 0)*.

Restul: #i = #j MOD #k

Codul operației este **MOD**.

Ca rezultat al operației, variabila #i va deveni restul raportului variabilelor #j şi #k. Valoarea lui #k nu poate fi 0, pentru că dacă este 0 se afişează mesajul de eroare *3092 DIVISION BY 0 # (împărțire prin 0)*.

Exemplu: Pentru #120 = 27 MOD 4, valoarea variabilei #120 va fi 3.

Produs logic, ŞI: #i = #j AND #k

Codul operației este **AND (ŞI logic)**.

Ca rezultat al operației, produsul logic dintre variabilele #j şi #k se va introduce în variabila #i pentru fiecare din cei 32 de biți. Se va introduce 1 atunci când pozițiile similare din cele două numere au valoarea 1 (altfel se introduce 0).

Funcții

Rădăcină pătrată: #i = SQRT #j

Codul operației este **SQRT**.

Ca rezultat al operației, variabila #i va deveni rădăcina pătrată a variabilei #j. Valoarea lui #j nu poate fi un număr negativ.

Sinus: #i = SIN #j

Codul operației este **SIN**.

Ca rezultat al operației, variabila #i va deveni sinusul variabilei #j. Valoarea lui #j se măsoară în grade.

Cosinus: #i = COS #j

Codul operației este **COS**.

Ca rezultat al operației, variabila #i va deveni cosinusul variabilei #j. Valoarea lui #j se măsoară în grade.

Tangenta: #i = TAN #j

Codul operației este **TAN**.

Ca rezultat al operației, variabila #i va deveni tangenta variabilei #j. Valoarea lui #j se măsoară în grade. Valoarea lui #j nu poate fi (2n+1)*90°, unde n=0, ±1, ±2, …

Arcsinus: #i = ASIN #j

Codul operației este **ASIN**.

Ca rezultat al operației, variabila #i va deveni arcsinus de variabila #j în grade. Condiția –1≤#j≤1 trebuie să fie adevărată. Rezultatul, adică valoarea lui #i este cuprins între +90° şi –90°.

Arccosinus: #i = AcIN #j

Codul operației este **ACOS**.

Ca rezultat al operației, variabila #i va deveni arccosinus de variabila #j în grade. Condiția $-1 \leq \#j \leq 1$ trebuie să fie adevărată. Rezultatul, adică valoarea lui #i este cuprins între 0° şi 180°.

Arctangentă: #i = ATAN #j

Codul operației este **ATAN**.

Ca rezultat al operației, variabila #i va deveni arctangentă de variabila #j în grade. Rezultatul, adică valoarea lui #i, este cuprins între +90° şi –90°.

Funcție exponențială cu baza e: #i = EXP #j

Codul operației este **EXP**.

Ca rezultat al operației, variabila #i va deveni e (numărul natural) la puterea #j.

Logaritm natural: #i = LN #j

Codul operației este **LN**.

Ca rezultat al operației, variabila #i va deveni logaritmul natural al numărului #j. Valoarea lui #j nu poate fi 0 sau un număr negativ.

Valoare absolută: #i = ABS #j

Codul operației este **ABS**.

Ca rezultat al operației, variabila #i va deveni valoarea absolută a variabilei #j.

Conversie din binar în zecimal codat binar: #i = BCD #j

Codul funcției este **BCD**.

Ca rezultat al operației, variabila #i va deveni valoarea BCD a variabilei #j. Valoarea variabilei #j este cuprinsă între 0 şi 99999999.

Conversie din zecimal codat binar în binar: #i = BIN #j

Codul funcției este **BIN**.

Ca rezultat al operației, variabila #i va deveni valoarea binară a variabilei #j. Valoarea variabilei #j este cuprinsă între 0 şi 99999999.

Ignorarea zecimalelor: #i = FIX #j

Codul funcției este **FIX**.

Această operație va ignora zecimalele variabilei #j şi acea valoare va trece în variabila #i.

De exemplu:

```
#130 = FIX 4.8 = 4
#131 = FIX –6.7 = -6
```

Rotunjirea zecimalelor prin majorare la 1: #i = FUP #j

Codul funcției este **FUP**.

Această operație va ignora zecimalele variabilei #j, dar va adăuga 1 la #j în valoare absolută.

De exemplu:

```
#130 = FUP 12.1 = 13
#131 = FUP –7.3 = -8
```

Operații aritmetice complexe – Succesiunea de executare

Operațiile aritmetice mai sus menționate se pot combina. Succesiunea de executare a operațiilor (ordinea de executare a operațiilor) este: funcții – operații de multiplicare – operații de sumare.
De exemplu:

```
#110 = #111 + #112 * COS  #113
                      ----1-----
              --------2-----------          ordinea operațiilor
      ------------3------------------
```

Modificarea succesiunii de executare

Succesiunea de executare a operațiilor poate fi modificată prin folosirea parantezelor **[** şi **]**. Pot fi până la cinci nivele de paranteze. Dacă într-un program se depăşeşte numărul de 5 nivele, se afişează eroarea *3064 BAD MACRO STATEMENT (declarație macro greşită)*.
Un exemplu de trei nivele de paranteze:

```
#120 = COS [ [ [ #121 - #122] * #123 + #125] * #126]
                 -----1---------
                 ------------2-----------
               -------------------3-----------------
           --------------------------4-----------------------
      ---------------------------5----------------------------
```

Numerele se referă la ordinea de executare a operațiilor. În interiorul unei paranteze este valabilă regula cu privire la ordinea de executare a operațiilor prezentată mai sus.

20.13.3 Operații logice

Limbajul de programare foloseşte următoarele operații logice:

egal	#i **EQ** #j
diferit	#i **NE** #j
mai mare	#i **GT** #j
mai mic	#i **LT** #j
mai mare sau egal	#i **GE** #j
mai mic sau egal	#i **LE** #j

Variabilele din ambele părți ale unei operații logice se pot înlocui şi cu formule. Expresiile condiționale de mai sus pot fi folosite în instrucțiuni de ramificare sau iterare: IF (dacă) respectiv WHILE (cât timp).
NOTĂ : Deoarece expresiile condiționale de mai sus sunt urmate de adunări şi scăderi, trebuie luate în considerație eventualele erori cu privire la acuratețea deciziei.

20.13.4 Ramificare necondiționată: GOTOn

Ca rezultat al instrucțiunii **GOTO**n, execuția programului se va continua în mod necondiționat la blocul cu numărul de secvență n din acelaşi program. Numărul de secvență n poate fi înlocuit cu o variabilă sau o formulă. Numărul blocului la care se face saltul prin instrucțiunea GOTO trebuie să fie pus la începutul blocului respectiv. Dacă nu este găsit numărul blocului selectat se afişează mesajul de eroare *3070 NOT EXISTING BLOCK NO.P (nu există bloc cu nr. p)*.

20.13.5 Ramificare condiționată: IF[<expresia condiției>] GOTOn

Dacă [<expresia condiției>], aflată obligatoriu între paranteze pătrate, este îndeplinită, execuția programului se va continua în mod necondiționat la blocul cu numărul de secvență n din acelaşi program.
Dacă [<expresia condiției>] nu este îndeplinită, execuția programului se va continua la blocul următor.
Se afişează mesajul de eroare *3091 ERRONEOUS OPERATION WITH # (operare eronată cu #)* în cazul în care IF nu este urmat de expresia condiției. Dacă expresia condiției are o eroare de sintaxă, se afişează mesajul de eroare *3064 BAD MACRO STATEMENT (declarație macro greşită)*.

20.13.6 Instrucțiune condiționată: IF[<expresia condiției>] THEN

Dacă (IF) [<expresia condiției>] este îndeplinită, va fi executată instrucțiunea care urmează după THEN (atunci).
Dacă [<expresia condiției>] nu este îndeplinită, execuția programului se va continua la blocul următor.
Cuvântul THEN se poate omite, instrucțiunea:

IF[<expresia condiției>] instrucțiune

va fi executată cu acelaşi rezultat.

20.13.7 Iterație: WHILE[<expresia condiției>] DOm … ENDm

Atât timp cât (WHILE) [<expresia condiției>] este îndeplinită, blocurile care urmează DOm (execută m) până la blocul ENDm (sfârşit m) vor fi executate repetat. În instrucțiune, comanda numerică va verifica dacă este îndeplinită condiția; dacă este, porțiunea de program cuprinsă între DOm şi ENDm va fi executată; apoi ca rezultat al instrucțiunii ENDm, programul se va întoarce pentru a verifica din nou expresia condiție de după WHILE.
Dacă [<expresia condiției>] nu este îndeplinită, execuția programului se va continua la blocul de după ENDm.
Dacă se omite WHILE[<expresia condiției>], adică ciclul este descris de instrucțiunea DOm …ENDm, porțiunea de program cuprinsă între DOm şi ENDm va fi executată la nesfârşit.
Valorile posibile pentru m sunt 1, 2, 3. Dacă se specifică orice altă valoare se afişează mesajul de eroare *3091 ERRONEOUS OPERATION WITH # (operare eronată cu #)*. Dacă WHILE nu este urmat de o expresie de condiție se afişează mesajul de eroare *3091 ERRONEOUS OPERATION WITH # (operare eronată cu #)*. Dacă expresia condiției are o eroare de sintaxă, se afişează mesajul de eroare *3064 BAD MACRO STATEMENT (declarație macro greşită)*.

Regulile de organizare a ciclului:
- Instrucțiunea DOm trebuie să fie specificată înaintea instrucțiunii ENDm.

```
:
END1
:
:            fals (EROARE 72)
:
DO1
```

- Instrucțiunile DOm şi ENDm trebuie să fie în perechi.

```
:
DO1
:
DO1          fals
:
END1
:

sau

:
DO1
:
END1         fals
:
END1
:
```

- Se poate folosi un anumit identificator de mai multe ori.

```
:
DO1
:
END1
:
:            corect
:
DO1
:
END1
:
```

- Perechile DOm … ENDm pot cuprinde până la trei nivele de alte perechi.

```
:
DO1
:
DO2
:
DO3
:
:            corect
:
END3
:
END3
:
END1
:
```

- Perechile DOm … ENDm nu se pot întrepătrunde.

```
:
DO1
:
DO2
:
:            fals
:
END1
:
END2
```

- Se poate face o ramificație în afara ciclului.

```
:
DO1
:
GOTO150
:
:            corect
:
END1
:
N150
:
```

- Nu este permisă intrarea în ciclu din afară.

```
:
GOTO150
:
DO1
:
:            fals
:
N150
:
END1
:
```

sau

```
:
DO1
:
N150
:
:            fals
:
END1
:
GOTO150
:
```

- Un subprogram sau un macro poate fi apelat din interiorul unui ciclu. Ciclurile din interiorul unui subprogram sau a unui macro utilizator pot la rândul lor să cuprindă până la trei nivele de alte cicluri.

```
:
DO1
:
M98…         corect
:
G65…         corect
:
G66…         corect
:
```

```
G67…        corect
:
END1
:
```

20.13.8 Comenzi de ieşire date

Comanda numerică va recunoaşte următoarele comenzi de ieşire date:
POPEN periferic deschis
BPRNT tipărire date binare (ieşire)
DPRNT tipărire date zecimale (ieşire)
PCLOS periferic închis
Aceste comenzi de ieşire date pot fi folosite pentru înscrierea externă a caracterelor şi valorilor variabilelor. Ieşirea se poate realiza în memoria comenzii numerice sau către un dispozitiv extern de stocare (printr-un canal serial).

Deschiderea unui periferic – POPENn
Înainte de a da o comandă de ieşire date, trebuie deschis perifericul corespunzător, prin care să se execute ieşirea datelor. Selectarea perifericul corespunzător se face prin numărul n:
n = 1 interfaţă cu canal serial RS-232C
n =31 memoria comenzii numerice
Un caracter % este de asemenea transmis către periferic simultan cu deschiderea perifericului, adică orice transmitere de date începe cu un caracter %.

Ieşire de date binare – BPRNT[…]

```
BPRNT [ a  #b  [c]  …  ]
                   unde  a   = caracter
                         #b = variabilă
                         [c] = numărul de cifre de după punctul zecimal
```

Comanda numerică va trimite caracterele în cod ISO sau ASCII (funcţie de impunerea parametrilor); variabilele vor fi transmise în formă binară.
- Caracterele vor fi transmise în cod ISO sau ASCII. Caracterele de transmis sunt:
caractere alfabetice (A, B, …, Z)
numere (1, 2, …,0)
caractere speciale (*, /, +, -)
Comanda numerică va trimite în locul codului ISO pentru * codul pentru spaţiu (A0h).
- Valorile variabilelor vor fi transmise în 4 bytes (adică în 32 de biţi), începând cu cel mai semnificativ byte. Numărul variabilei trebuie urmat de numărul de zecimale între paranteze pătrate. Comanda numerică va converti valoarea în virgulă mobilă a variabilei în valoare cu virgulă fixă, în care numărul de cifre zecimale semnificative este egal cu valoarea pusă între parantezele pătrate. Valorile posibile pentru c sunt 1, 2, …, 8.
Dacă, de exemplu # 120 = 258.647673 şi [3] -------- va fi transmis 258648 = 0003F258h
- O variabilă vacantă va fi transmisă cu codul binar 00000000h.
- La sfârşitul transmisiei de date, comanda numerică va trimite automat caracterul **L**ine **F**eed (sfârşit de rând).
De exemplu:

```
BPRNT [ C*/ X#110 [3] Y#120 [3] M#112 [0] ]
      #110=318.49362  ------------------   318494=0004DC1Eh
      #120=0.723415   ------------------   723=000002D3h
      #112=23.9       ------------------   24=00000018h
```

Caracterele de transmis sunt:

```
7 6 5 4 3 2 1 0

1 1 0 0 0 0 1 1 ---C
1 0 1 0 0 0 0 0 ---Spaţiu
1 0 1 0 1 1 1 1 ---/
1 1 0 1 1 0 0 0 ---X
0 0 0 0 0 0 0 0 ---00
0 0 0 0 0 1 0 0 ---04
1 1 0 1 1 1 0 0 ---DC
0 0 0 1 1 1 1 0 ---1E
0 1 0 1 1 0 0 1 ---Y
0 0 0 0 0 0 0 0 ---00
0 0 0 0 0 0 0 0 ---00
0 0 0 0 0 0 1 0 ---02
1 1 0 1 0 0 1 1 ---D3
0 1 0 0 1 1 0 1 ---M
0 0 0 0 0 0 0 0 ---00
0 0 0 0 0 0 0 0 ---00
0 0 0 0 0 0 0 0 ---00
0 0 0 1 1 0 0 0 ---18
0 0 0 0 1 0 1 0 ---Line Feed(sfârşit de rând)
```

Ieşire de date zecimale – DPRNT[…]

```
DPRNT [ a  #b  [ c d ]  …  ]
                  unde a   = caracter
                       #b = variabilă
                       d  = număr de cifre după punctul zecimal
                       c  = număr de cifre înainte de punctul zecimal
```

Toate caracterele şi cifrele vor fi trimise în cod ISO sau ASCII, funcţie de impunerea parametrilor.
 - Pentru regulile de trimitere a caracterelor vezi instrucţiunile de la **BPRNT**.
 - Pentru ieşirea valorilor variabilelor, trebuie precizate numerele întregilor zecimali şi ale fracţiilor, în care trebuie pusă variabila. Cifrele trebuie specificate între paranteze pătrate. Pentru specificarea cifrelor trebuie îndeplinită condiţia $0 < c + d < 9$. Procedura de transmitere la ieşire a cifrelor începe cu cea mai semnificativă cifră. La transmiterea la ieşire a cifrelor, semnul negativ (-) şi punctul zecimal (.) vor fi transmise folosindu-se codurile ISO respective. Dacă parametrul PRNT=0, pentru semnul + şi pentru zerourile din faţa întregilor se va transmite un cod de spaţiu; după punctul zecimal fiecare 0 (dacă există) se va transmite cu codul de 0. Dacă parametrul PRNT=1, semnul + şi zerourile din faţa întregilor nu vor fi transmise; dacă este definit punctul zecimal, zerourile aflate după el vor fi transmise. În caz contrar nu se vor transmite nici punctul zecimal nici zerourile.
 - Dacă d=0, va fi transmis punctul zecimal; dacă este specificat doar c, nu va fi transmis nici punctul zecimal.
 - O variabilă vacantă va fi transmisă cu codul 0.

- La sfârşitul transmisiei de date, comanda numerică va trimite automat caracterul **L**ine **F**eed (sfârşit de rând).

De exemplu:

```
DPRNT [ X#130 [53] Y#500 [53] T#10 [2] ]
      #130=35.897421    -------- 35.897
      #500=-150.8       -------- -150.8
      #10=214.8         -------- 15
```

Ieşirea de date cu PRNT=0

```
7 6 5 4 3 2 1 0

1 1 0 1 1 0 0 0 ---X
1 0 1 0 0 0 0 0 ---Spaţiu
1 0 1 0 0 0 0 0 ---Spaţiu
1 0 1 0 0 0 0 0 ---Spaţiu
1 0 1 0 0 0 0 0 ---Spaţiu
0 0 1 1 0 0 1 1 ---3
0 0 1 1 0 1 0 1 ---5
0 0 1 0 1 1 1 0 ---Punct zecimal (.)
1 0 1 1 1 0 0 0 ---8
0 0 1 1 1 0 0 1 ---9
1 0 1 1 0 1 1 1 ---7
0 1 0 1 1 0 0 1 ---Y
0 0 1 0 1 1 0 1 ---Semnul minus (-)
1 0 1 0 0 0 0 0 ---Spaţiu
1 0 1 0 0 0 0 0 ---Spaţiu
1 0 1 1 0 0 0 1 ---1
0 0 1 1 0 1 0 1 ---5
0 0 1 1 0 0 0 0 ---0
0 0 1 0 1 1 1 0 ---Punct zecimal (.)
1 0 1 1 1 0 0 0 ---8
0 0 1 1 0 0 0 0 ---0
0 0 1 1 0 0 0 0 ---0
1 1 0 1 0 1 0 0 ---T
1 0 1 0 0 0 0 0 ---Spaţiu
1 0 1 1 0 0 0 1 ---1
0 0 1 1 0 1 0 1 ---5
0 0 0 0 1 0 1 0 ---Line Feed (LF)(sfârşit de rând)
```

Ieşirea de date cu PRNT=1

```
7 6 5 4 3 2 1 0

1 1 0 1 1 0 0 0 ---X
0 0 1 1 0 0 1 1 ---3
0 0 1 1 0 1 0 1 ---5
0 0 1 0 1 1 1 0 ---Punct zecimal (.)
1 0 1 1 1 0 0 0 ---8
0 0 1 1 1 0 0 1 ---9
1 0 1 1 0 1 1 1 ---7
0 1 0 1 1 0 0 1 ---Y
0 0 1 0 1 1 0 1 ---Semnul minus (-)
1 0 1 1 0 0 0 1 ---1
0 0 1 1 0 1 0 1 ---5
0 0 1 1 0 0 0 0 ---0
0 0 1 0 1 1 1 0 ---Punct zecimal (.)
1 0 1 1 1 0 0 0 ---8
0 0 1 1 0 0 0 0 ---0
0 0 1 1 0 0 0 0 ---0
```

```
1 1 0 1 0 1 0 0 ---T
1 0 1 1 0 0 0 1 ---1
0 0 1 1 0 1 0 1 ---5
0 0 0 0 1 0 1 0 ---Line Feed (LF)(sfârşit de rând)
```

Închiderea unui periferic - PCLOSn

Perifericul deschis cu comanda POPEN trebuie să fie închis cu comanda PCLOS. Comanda PCLOS trebuie să fie urmată de o specificare a numărului perifericului care să fie închis. În momentul închiderii, este trimis la periferic şi un caracter %, adică fiecare transmisie de date este încheiată cu un caracter %.

Note:

- Secvenţa de ieşire date este fixă. La început trebuie deschis perifericul corespunzător cu comanda POPEN, apoi se transmit datele (cu una din comenzile BPRNT sau DPRNT), iar în final perifericul deschis trebuie să fie închis cu instrucţiunea PCLOS.
- Deschiderea şi închiderea unui periferic pot fi specificate în orice punct din program. De exemplu, poate fi deschis la începutul programului şi închis la sfârşitul programului, datele fiind transmise oricând în timpul programului.
- O comandă M30 sau M2 executată în timpul în care se execută transmiterea de date va întrerupe transferul. Pentru a se evita aceasta, trebuie introdusă o perioadă de aşteptare înainte de executarea comenzii M30.
- Parametrii perifericului (rata baud, numărul de biţi de stop, etc.) trebuie impuşi corect. Aceştia pot fi selectaţi în grupul SERIAL din câmpul parametrilor.

20.14 Instrucţiuni NC şi instrucţiuni macro

Blocurile NC şi blocurile macro pot fi diferenţiate în limbajul de programare. Blocurile scrise în termenii codurilor convenţionale G, M, etc. sunt privite ca blocuri NC chiar şi atunci când valorilor adreselor li se atribuie variabile sau formule la fel ca şi valori numerice. Blocurile următoare sunt privite ca instrucţiuni macro:

- un bloc care conţine o instrucţiune de definiţie sau de substituţie (#i=#j)
- un bloc care conţine o ramificare condiţionată sau o instrucţiune de iteraţie (IF, WHILE)
- blocuri care conţin comenzi de control (GOTO, DO, END)
- blocuri care conţin apeluri macro (G65, G66,G66.1, G67, sau coduri G, sau M care iniţiază apeluri macro).

20.15 Desfăşurarea în timp a execuţiei instrucţiunilor NC şi macro

Blocurile macro pot fi executate de către comanda numerică în paralel cu blocurile NC sau în ordine consecutivă. Parametrul SBSTM determină executarea blocurilor NC şi macro. Dacă parametrul:

=0: blocurile NC şi cele macro sunt executate în ordinea în care au fost scrie în program,

=1: Declaraţiile macro sunt executate în decursul executării blocurilor NC.

Exemplu:

SBSTM=0

```
%O1000
...
N10 #100=50
N20 #101=100
N30 G1 X#100 Y#101
N40 #100=60 (definire după N30)
N50 #101=120 (definire după N30)
N60 G1 X#100 Y#101
```

Comenzile de definire din blocurile N40 şi N50 sunt executate după executarea mişcării din blocul N30.

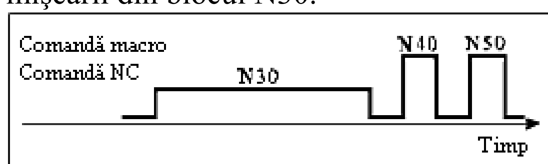


Fig. 20.15-1

Concluzii:
- *execuţia programului este mai lentă,*
- *dacă execuţia blocului N30 este întreruptă şi apoi prelucrarea este pornită, prelucrarea continuă deoarece variabilele blocului N30 nu au fost modificate de către blocurile N40, N50.*

SBSTM=1

```
%O1000
...
N10 #100=50
N20 #101=100
N30 G1 X#100 Y#101
N40 #100=60 (definire în timpul N30)
N50 #101=120 (definire în timpul N30)
N60 G1 X#100 Y#101
```

Comenzile de definire din blocurile N40 şi N50 sunt executate în timpul executării mişcării din blocul N30.

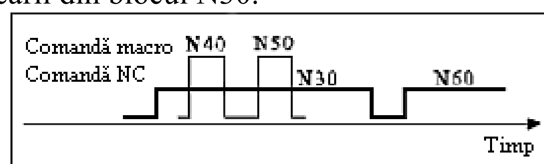


Fig. 20.15-2

Concluzii:
- *execuţia programului este mai rapidă,*
- *dacă execuţia blocului N30 este întreruptă şi apoi prelucrarea este pornită, prelucrarea nu poate continua, decât dacă se dă căutare de bloc pentru blocul N30 deoarece variabilele blocului N30 sunt deja modificate de către blocurile N40, N50.*

20.16 Afişarea macro şi a subprogramelor în modul automat

Blocurile de macro şi subprogramele vor fi afişate de către comanda numerică în modul automat. Dacă parametrul *MD8* este impus 0, blocurile de subprogram şi macro numerotate de la 8000 la 8999 nu vor fi listate atunci când sunt executate. Dacă parametrul *MD8* este impus 1, vor fi listate şi aceste blocuri.
Dacă parametrul *MD9* este impus 0, blocurile de subprogram şi macro numerotate de la 9000 la 9999 nu vor fi listate atunci când sunt executate. Dacă parametrul *MD9* este impus 1, vor fi listate şi aceste blocuri.

20.17 Folosirea butonului STOP în timpul execuţiei unei instrucţiuni macro

Apăsarea butonului STOP, adică suspendarea execuţiei programului va avea efect întotdeauna la terminarea instrucţiunii macro aflate în execuţie.

20.18 Ciclul macro de frezare în adâncime

Instrucțiunea

G65 P9999 X Y Z I J K R F D E Q M S T

va porni un ciclu de frezare în adâncime. Pentru executarea ciclului, macro cu număr de program O9999 trebuie adus în memorie din memoria PROM a comenzii numerice. Anterior apelării ciclului, scula trebuie adusă deasupra centrului geometric al prelucrării în planul selectat, la o distanță de siguranță deasupra piesei. La sfârşitul ciclului scula va fi retrasa în acelaşi punct.

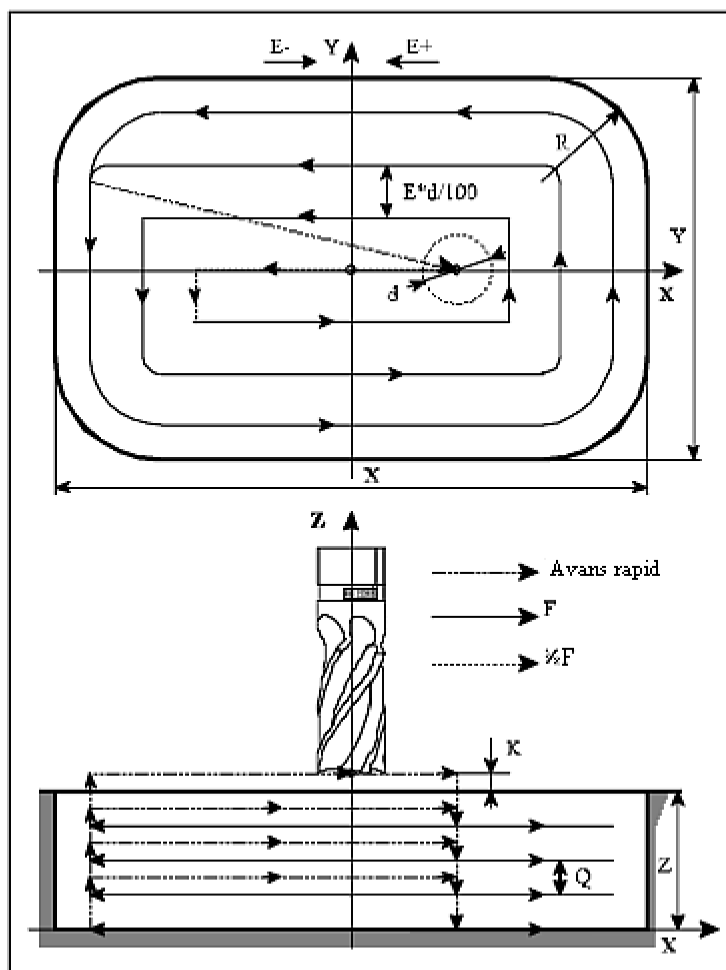


Fig. 20.18-1

Adresele în bloc trebuie să aibă următoarea semnificație:

X = mărimea prelucrării în direcția X

Y = mărimea prelucrării în direcția Y

Z = mărimea prelucrării în direcția Z

Instrucțiunile G17, G18, G19 vor defini lungimea, lățimea şi adâncimea prelucrării pentru cele trei coordonate. De exemplu, în cazul lui G17, Z va fi adâncimea prelucrării, cea mai lungă dintre X şi Y va fi lungimea prelucrării, iar cea mai scurtă lățimea. Aceste valori trebuie să fie introduse în valori absolute ca numere pozitive.

R = raza colțurilor prelucrării. Rotunjirea colțurilor prelucrării (dacă există) trebuie specificată la adresa R. Dacă nu se completează adresa R, colțurile prelucrării se vor rotunji cu raza sculei.

I = distanța de siguranță pe direcția adâncimii prelucrării în cazul lui G19.

J = distanța de siguranță pe direcția adâncimii prelucrării în cazul lui G18.

K = distanța de siguranță pe direcția adâncimii prelucrării în cazul lui G17.

În funcție de planul selectat, distanța de siguranță pe direcția sculei trebuie să fie specificată în bloc la adresele I(G19), J(G18) sau K(G17). Când este pornit ciclul, comanda numerică consideră că vârful sculei este localizat la acea distanță față de suprafața piesei. În timpul prelucrării, după ce s-a îndepărtat materialul de la un nivel, scula va fi ridicată la acea distanță astfel încât să fie adusă în punctul de pornire pentru frezarea nivelului următor.

D = adresa registrului care conține compensarea razei sculei.

Numărul de compensare al razei sculei folosite în program trebuie specificat obligatoriu la adresa D. În plus, frezarea în adâncime trebuie să fie executată în starea G40.

E = lățime de tăiere, ca procent din diametrul de frezare
cu semnul +, prelucrare în sens antiorar,
cu semnul -, prelucrare în sens orar.
La adresa E se pot specifica două tipuri de informație. Valoarea lui E defineşte adâncimea de tăiere ca procent din diametrul de frezare. Dacă nu este specificată, comanda numerică va considera în mod automat +83%. Comanda numerică poate modifica data specificată la adresa E în funcție de adâncimea prelucrării, pentru a se obține o tăiere uniformă la frezarea unui anumit nivel. O astfel de modificare poate să fie doar o reducere. Semnul adresei E defineşte direcția de frezare. Atunci când avem E+, adică pozitiv, prelucrarea se va executa în sens antiorar, iar dacă avem E-, adică negativ, prelucrarea se va executa în sens orar.
Q = adâncimea de tăiere
Adâncimea de tăiere poate fi specificată la adresa Q în unitatea de măsură curentă (mm sau inci). În funcție de adâncimea prelucrării, comanda numerică poate să nu țină cont de valoarea din program cu scopul de a obține o distribuție uniformă a tăieturilor. O astfel de modificare poate să fie doar o reducere.
F = avansul
Avansul aplicat în ciclu poate fi specificat la adresa F. Dacă nu se precizează o valoare pentru F, va fi adoptată valoarea modală pentru F.
Se va aplica 50% din F
- la începutul frezării unui nivel, şi la găurirea unei adâncimi de tăiere (Q),
- când este încărcată în ambele părți frezarea prelucrării longitudinal pe o distanță egală cu Q.

M S T = funcție
O funcție M, S, T poate fi specificată în blocul care apelează procedura de frezare în adâncime, funcție care va fi executată de către comanda numerică înainte de începerea frezării.

Cazuri particulare de frezare în adâncime:

Dacă nu s-a specificat lățimea prelucrării, se va considera ca lățime a prelucrării de două ori raza colțurilor prelucrării.

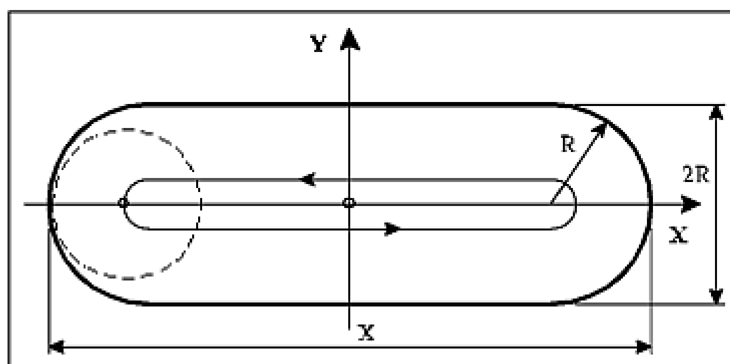


Fig. 20.18-2

Dacă nu s-au specificat lățimea prelucrării şi razele de rotunjire a colțurilor, se va considera ca lățime diametrul sculei folosite (canal).

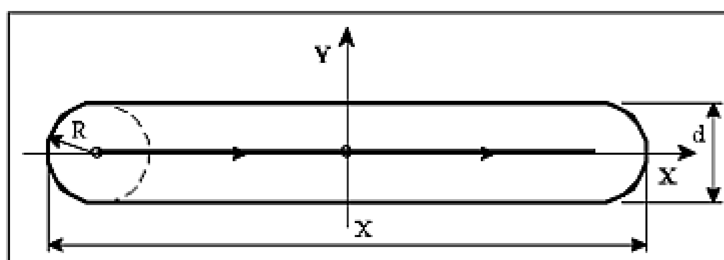


Fig. 20.18-3

Dacă nu s-au specificat nici lungimea nici lățimea prelucrării, ci s-a programat numai adresa R, se va obține o frezare circulară de rază R.

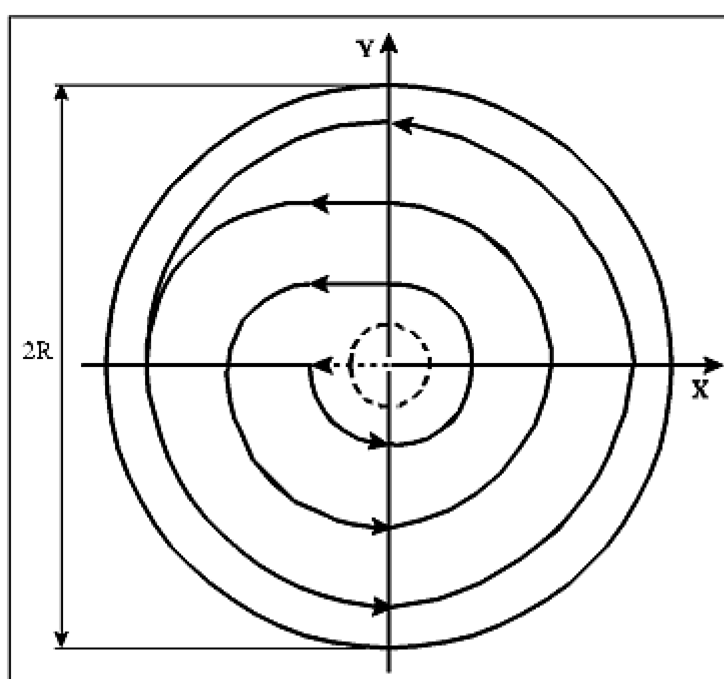


Fig. 20.18-4

Dacă nu sunt specificate nici lungimea nici lățimea nici raza, ciclul va degenera în găurire.

Mesaje de eroare în cursul frezări în adâncime:

MACRO ERROR 1 (eroare macro 1) – specificare falsă a blocului. Cauze posibile:
- adâncimea prelucrării nespecificată
- raza sculei nespecificată
- adâncimea de tăiere nespecificată

MACRO ERROR 2 (eroare macro 2) – eroare de definire în dimensiunile specificate. Cauze posibile:

- mărimea specificată pentru lungimea sau lățimea prelucrării este mai mică decât dublul razei prelucrării.
- lungimea sau lățimea prelucrării este mai mică decât diametrul sculei apelate la adresa D.
- valoarea specificată pentru lățimea de tăiere este 0 sau raza sculei apelate este 0.
- valoarea adâncimii de tăiere este 0, adică s-a programat 0 la adresa Q.

Note

Index în ordine alfabetică